Web Application Scanning API

User Guide

Version 4.2

July 6, 2015

CONTINUOUS SECURITY

Copyright 2012-2015 by Qualys, Inc. All Rights Reserved.

Qualys, the Qualys logo and QualysGuard are registered trademarks of Qualys, Inc. All other trademarks are the property of their respective owners.

Qualys, Inc.
1600 Bridge Parkway
Redwood Shores, CA 94065
1 (650) 801 6100

CONTENTS

Preface

Chapter 1 Welcome

Get Started 9
Get API Notifications 9
Introduction to the WAS API Paradigm 10
Base URL to the Qualys API Server 16
How to Download Vulnerability Details 17

Chapter 2 Web Application API

Current web application count 22
Search web applications 25
Get details for a web application 29
Create a web application 33
Update a web application 47
Delete web applications 53
Purge web applications 56
Reference: WebApp 59

Chapter 3 Authentication API

Current authentication record count 66
Search authentication records 69
Get details for an authentication record 72
Create a new authentication record 75
Update an authentication record 83
Delete authentication records 85

Chapter 4 Scan API

Current scan count 90
Search scans 94
Get scan details 100
Launch a new scan 104
Retrieve the status of a scan 109
Retrieve the results of a scan 110
Cancel an unfinished scan 117
Delete an existing scan 118
Reference: WasScan 121
Reference: WAS Scan Results (legacy) 124

Chapter 5 Schedule API

Current schedule count ... 128
Search schedules ... 131
Get schedule details ... 137
Create a schedule ... 140
Update a schedule ... 148
Activate an existing schedule ... 151
Deactivate an existing schedule ... 153
Delete one or more existing schedules ... 155
Download one or more schedules to iCalendar ... 158
Reference: WasScanSchedule ... 163

Chapter 6 Report API

Current report count ... 170
Search reports ... 172
Get report details ... 177
Get report status ... 179
Download a report ... 180
Send an encrypted PDF report ... 181
Update a report ... 183
Delete one or more existing reports ... 185
Reference: Report ... 188

Chapter 7 Report Creation API

Report Creation API ... 192
Web Application Report ... 194
Scan Report ... 198
Scorecard Report ... 202
Catalog Report ... 206
Reference: Report Creation ... 209

Chapter 8 Option Profile API

Current option profile count ... 214
Search option profiles ... 217
Get details for an option profile ... 220
Create a new option profile ... 223
Update an option profile ... 230
Delete an option profile ... 234

Chapter 9 Finding API

Current finding count ... 238
Search findings ... 241
Get details of a finding ... 249
Ignore Findings ... 254
Activate Findings ... 256

Chapter 10 Progressive Scanning

Web Application API 260
Scan API 263
Schedule API 266
Scan Report 269

Appendix A Error Messages

Appendix B WAS Findings in XML Reports

Preface

Using the Qualys Web Application Scanning (WAS) API, third parties can integrate the Qualys Security and Compliance solution into their own applications using an extensible XML interface. This user guide is intended for application developers who will use the Qualys WAS API.

About Qualys

Qualys, Inc. (NASDAQ: QLYS) is a pioneer and leading provider of cloud security and compliance solutions with over 7,700 customers in more than 100 countries, including a majority of each of the Forbes Global 100 and Fortune 100.The Qualys Cloud Platform and integrated suite of solutions help organizations simplify security operations and lower the cost of compliance by delivering critical security intelligence on demand and automating the full spectrum of auditing, compliance and protection for IT systems and web applications. Founded in 1999, Qualys has established strategic partnerships with leading managed service providers and consulting organizations including Accenture, Accuvant, BT, Cognizant Technology Solutions, Dell SecureWorks, Fujitsu, HCL Comnet, InfoSys, NTT, Tata Communications, Verizon and Wipro. The company is also a founding member of the Cloud Security Alliance (CSA).

For more information, please visit www.qualys.com.

Contact Qualys Support

Qualys is committed to providing you with the most thorough support. Through online documentation, telephone help, and direct email support, Qualys ensures that your questions will be answered in the fastest time possible. We support you 7 days a week, 24 hours a day. Access support information at www.qualys.com/support/.

CHAPTER 1

Welcome

Welcome to Qualys Web Application Scanning API. Several functional suites are available to support WAS scanning and reporting.

Get Started

Introduction to the WAS API Paradigm - Review important information about the WAS API framework.

Base URL to the Qualys API Server - Learn the basics about making API requests. The base URL depends on the platform where your Qualys account is located.

- We'll tell you about the method used for authentication. API requests must authenticate using Qualys credentials.

How to Download Vulnerability Details - We'll walk you through the steps, using the KnowledgeBase API. You can download vulnerability descriptions and recommended fixes.

Get API Notifications

We recommend you join our Community and subscribe to our API notifications so you'll get email notifications telling you about important upcoming API enhancements and changes.

From our Community

Join our Community

Subscribe to API Notifications (select Receive email notifications)

Introduction to the WAS API Paradigm

The new Qualys WAS API framework introduces numerous innovations and new functionality compared to the other Qualys API frameworks.

Request URL

The URL for making API requests respects the following structure:

https://<baseurl>/qps/rest/3.0/<operation>/<module>/<object>/<object_id>

where the components are described below.

<baseurl>	The Qualys API server URL that you should use for API requests depends on the platform where your account is located. The base URL for Qualys US Platform 1 is: https://qualysapi.qualys.com
<operation>	The request operation, such as get a list, get a count, search, create, and update.
<module>	The API module. For the WAS API, the module is: “was”.
<object>	The module specific object.
<object_id>	(Optional) The module specific object ID, if appropriate.

Making Requests with an XML Payload

While it is still possible to create simple API requests using the GET method, you can create API requests using the POST method with an XML payload to make an advanced request.

The XML payloads can be compared to a scripting language that allows user to make multiple actions within one single API request, like adding a parameter to an object and updating another parameter.

The XML structure of the payload is described in the XSD files.

Using Curl

Curl is a multi-platform command-line tool used to transfer data using multiple protocols. This tool is supported on manu systems, including Windows, Unix, Linux and Mac. In this document Curl is used in the examples to build WAS API requests using the HTTP over SSL (https) protocol, which i s required by the Qualys WAS API framework.

Want to learn more? Visit http://curl/haxx/se

The following Curl options are used according to different situations:

Option	Description
-u “LOGIN:PASSWORD”	This option is used for basic authentication.
-X “POST”	This option is used to provide a method other than the default method, GET.
-H “content-type”	This option is used to provide a custom HTTP request header parameter for content type, to specify the MIME type of the curl’s payload.
--data-binary	This option is used to specify the POST data. See the examples below.

The sample below shows a typical Curl request using options mentioned above and how they interact with each other. The option -X “POST” tells Curl to execute the request using the HTTP POST method. The option “--data-binary @-” tells Curl to read the POST data from its standard input (stdin). The string “< file.xml” is interpreted by the shell to redirect the content of the file to the stdin of the command. The option -H “content-type: text/xml” tells Curl the POST data in “file.xml” is XML in text format.

curl -H “content-type: text/xml” -X “POST” --data-binary @- “https://example.com” < file.xml

This documentation uses Curl examples showing the POST data in the “file.xml” file. This is referred to as Request POST Data. This can also be referred to as the Payload.

XML Output and Schemas

Web Application XSD https://qualysapi.qualys.com/qps/xsd/3.0/was/webapp.xsd
Authentication XSD https://qualysapi.qualys.com/qps/xsd/3.0/was/webappauthrecord.xsd
Scan XSD (both valid) https://qualysapi.qualys.com/qps/xsd/3.0/was/scan.xsd https://qualysapi.qualys.com/qps/xsd/3.0/was/wasscan.xsd
Schedule XSD (both valid) https://qualysapi.qualys.com/qps/xsd/3.0/was/schedule.xsd https://qualysapi.qualys.com/qps/xsd/3.0/was/wasscanschedule.xsd
Report XSD https://qualysapi.qualys.com/qps/xsd/3.0/was/report.xsd
Option Profile XSD https://qualysapi.qualys.com/qps/xsd/3.0/was/optionprofile.xsd
Finding XSD https://qualysapi.qualys.com/qps/xsd/3.0/was/finding.xsd

XML Output Pagination / Truncation Logic

The XML output of a search API request is paginated and the default page size is 100 object records. The page size can be customized to a value between 1 and 1,000. If the number of records is greater than the page size then the <ServiceResponse> element shows the response code SUCCESS with the element <hasMoreRecords>true</hasMoreRecords> as shown below.

Follow the process below to obtain the first two the XML pages for an API request. Please apply the same logic to get all the next (n+1) pages until all records are returned. This is indicated when <hasMoreRecords>false</hasMoreRecords>.

Request 1:

Search for web applications that have a name containing the string “Merchant”. The service request in the POST data file “file.xml” defines this search critera.

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/webapp" <
file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data for Request 1:

```
<ServiceRequest>
  <preferences>
    <limitResults>5</limitResults>
  </preferences>
  <filters>
    <Criteria field="name" operator="CONTAINS">Merchant</Criteria>
  </filters>
</ServiceRequest>
```

Response:

The number of records is greater than the default pagination value so the <ServiceResponse> element identifies the last ID of the object in the current page output.

```
<ServiceResponse ...>
   <responseCode>SUCCESS</responseCode>
   <COUNT>5</COUNT>
   <hasMoreRecords>true</hasMoreRecords>
   <lastId>123</lastId>
   <data>
      <!--here you will find 5 web application records-->
   </data>
</ServiceResponse>
```

Request 2:

To get the next page of results, you need to edit your service request in "file.xml" that will be passed to API request as a POST payload. According to the <lastId> element returned in the first page, you want the next page of results to start with the object ID 124 or greater.

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/webapp" <
file.xml
```

Request POST Data for Request 2:

You'll notice the operator field value is set to 123, which is the value returned in <lastId> of the previous page output. The GREATER operator is a logical "greater than" (it does not mean greater than or equal to).

```
<ServiceRequest>
   <filters>
      <Criteria field="name"
```

```
operator="CONTAINS">Merchant</Criteria>
      <Criteria field="id" operator="GREATER">123</Criteria>
   </filters>
</ServiceRequest>
```

Setting the Custom Page Size

The service request needs to contain the <preferences> section with the <limitResults> parameter. For the <limitResults> parameter you can enter a value from 1 to 1,000.

```
<ServiceRequest>
  <filters>
    <Criteria> ... </Criteria>
  </filters>
  <preferences>
    <limitResults>200</limitResults>
  </preferences>
</ServiceRequest>
```

Authentication

The application must authenticate using Qualys account credentials (user name and password) as part of the HTTP request. The credentials are transmitted using the "Basic Authentication Scheme" over HTTPS.

For more information, see the "Basic Authentication Scheme" section of RFC #2617:

```
http://www.faqs.org/rfcs/rfc2617.html
```

The exact method of implementing authentication will vary according to which programming language is used.

The allowed methods, POST and/or GET, for each API request are documented with each API call in this user guide.

Basic authentication - recommended option:

```
curl -u "USERNAME:PASSWORD"
https://qualysapi.qualys.com/qps/rest/3.0/count/was/webapp
```

where qualysapi.qualys.com is the base URL to the Qualys API server where your account is located.

Base URL to the Qualys API Server

The Qualys API documentation and sample code within it use the API server URL for Qualys US Platform 1: qualysapi.qualys.com.

The Qualys API server URL that you should use for API requests depends on the platform where your account is located.

Account Location	API Server URL
Qualys US Platform 1	https://qualysapi.qualys.com
Qualys US Platform 2	https://qualysapi.qg2.apps.qualys.com
Qualys EU Platform	https://qualysapi.qualys.eu

How to Download Vulnerability Details

When you download web application scan results using the WAS API, you'll want to view vulnerability descriptions from the Qualys KnowledgeBase in order to understand the vulnerabilities detected and see our recommended solutions. You can do this programmatically using the "KnowledgeBase" API v2 (api/2.0/fo/knowledge_base/vuln/?action=list). This API function is part of Qualys API v2 and it's described in the Qualys API v2 User Guide (click here to download the latest version of the API v2 User Guide).

Making API Requests

Authentication with valid Qualys credentials is required for making Qualys API requests. When calling the V2 API functions, you have the option to choose: 1) session based authentication, using login and logout operations, or 2) basic HTTP authentication.

The GET or POST access method may be used to make an API request.

Authorized Qualys users have permissions to download vulnerability data using the KnowledgeBase API V2. Please contact Qualys Support or your sales representative if you would like to obtain authorization for your subscription.

For further information, please refer to the Qualys API v2 User Guide.

Parameters

The input parameters for the KnowledgeBase API v2 are described below. Several optional input parameters may be specified. When unspecified, the XML output includes all vulnerabilities in the KnowledgeBase, showing basic details for each vulnerability. Several optional parameters allow you specify filters. When filter parameters are specified, these parameters are ANDed.

Parameter	Description
action=list	(Required) A flag used to request the download of vulnerability data from the KnowledgeBase.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
details={**Basic** | All | None}	(Optional) Show the requested amount of information for each vulnerability in the XML output. A valid value is: Basic (default), All, or None. Basic includes basic elements plus CVSS Base and Temporal scores. All includes all vulnerability details, including the Basic details.

Parameter	Description
ids={value}	(Optional) Used to filter the XML output to include only vulnerabilities that have QID numbers matching the QID numbers you specify.
id_min={value}	(Optional) Used to filter the XML output to show only vulnerabilities that have a QID number greater than or equal to a QID number you specify.
id_max={value}	(Optional) Used to filter the XML output to show only vulnerabilities that have a QID number less than or equal to a QID number you specify.
is_patchable={0|1}	(Optional) Used to filter the XML output to show only vulnerabilities that are patchable or not patchable. A vulnerability is considered patchable when a patch exists for it. When 1 is specified, only vulnerabilities that are patchable will be included in the output. When 0 is specified, only vulnerabilities that are not patchable will be included in the output. When unspecified, patchable and unpatchable vulnerabilities will be included in the output.
last_modified_after={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified after a certain date and time. When specified vulnerabilities last modified by a user or by the service will be shown. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
last_modified_before={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified before a certain date and time. When specified vulnerabilities last modified by a user or by the service will be shown. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
last_modified_by_user_after={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified by a user after a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
last_modified_by_user_before={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified by a user before a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).

Parameter	Description
last_modified_by_service_after={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified by the service after a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
last_modified_by_service_before={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified by the service before a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
published_after={date}	(Optional) Used to filter the XML output to show only vulnerabilities published after a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
published_before={date}	(Optional) Used to filter the XML output to show only vulnerabilities published before a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
discovery_method={value}	(Optional) Used to filter the XML output to show only vulnerabilities assigned a certain discovery method. A valid value is: Remote, Authenticated, RemoteOnly, AuthenticatedOnly, or RemoteAndAuthenticated. When "Authenticated" is specified, the service shows vulnerabilities that have at least one associated authentication type. Vulnerabilities that have at least one authentication type can be detected in two ways: 1) remotely without using authentication, and 2) using authentication.
discovery_auth_types={value}	(Optional) Used to filter the XML output to show only vulnerabilities having one or more authentication types. A valid value is: Windows, Oracle, Unix or SNMP. Multiple values are entered as a comma-separated list.
show_pci_reasons={**0** | 1}	(Optional) Used to filter the XML output to show reasons for passing or failing PCI compliance (when the CVSS Scoring feature is turned on in the user's subscription). Specify 1 to view the reasons in the XML output. When unspecified, the reasons are not included in the XML output.

Sample API Requests

These sample requests work on Qualys US Platform 1 where the FQDN in the API server URL is qualysapi.qualys.com. Please be sure to replace the FQDN with the proper API server URL for your platform. For the EU platform, use qualysapi.qualys.eu. For a partner platform, use the URL for your @customer platform API server.

Sample 1. Request all vulnerabilities in the KnowledgeBase showing basic details:

```
curl -k -u "user:password" -H "X-Requested-With: Curl" -X "POST"
-d "action=list"
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/" >
output.txt
```

Sample 2. Request patchable vulnerabilities that have QIDs 1-200 showing all details:

```
curl -k -u "user:password" -H "X-Requested-With: Curl" -X "POST"
-d "action=list&ids=1-200&is_patchable=1&details=All"
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/" >
output.txt
```

Sample 3. Request vulnerabilites that were last modified by the service after July 20, 2011 and that have the "remote and authenticated" discovery method:

```
curl -k -u "user:password" -H "X-Requested-With: Curl" -X "POST"
-d "action=list&last_modified_by_service_after=2011-07-20
&discovery_method=RemoteAndAuthenticated"
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/" >
output.txt
```

XML Output

A KnowledgeBase API request returns XML output using the knowledge_base_vuln_list_output.dtd, which can be found at the following URL (where qualysapi.qualys.com is your API server URL):

```
https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/knowle
dge_base_vuln_list_output.dtd
```

The DTD for the KnowledgeBase output is described in the Qualys API v2 User Guide, in Appendix A.

CHAPTER 2

Web Application API

The WAS Web Application API provides a suite of API functions for managing web applications that you want to scan for security risks.

These operations are available:

Current web application count

Search web applications

Get details for a web application

Create a web application

Update a web applicationn

Delete web applications

Purge web applications

Current web application count

Returns the total number of web applications in the user's account. Input elements are optional and are used to filter the number of web applications included in the count.

URL: https://qualysapi.qualys.com/qps/rest/3.0/count/was/webapp

Methods allowed: GET, POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: WebApp for descriptions of all <WebApp> elements.

id (Integer)	createdDate (Date)
name (Text)	updatedDate (Date)
url (Text)	isScheduled (Boolean)
tags.name (Text)	isScanned (Boolean)
tags.id (Integer)	lastScan.status (Keyword: SUBMITTED, RUNNING, FINISHED, ERROR or CANCELLED)
	lastScan.date (Date)

Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN
Boolean	(true/false) EQUALS, NOT EQUALS

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Count includes web applications within the user's scope

Examples

Example 1: Count - no criteria (GET)

Get the number of web applications in the user's account.

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/webapp"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>227</count>
</ServiceResponse>
```

Example 2: Count - criteria (POST)

Get the number of web applications in the user's account, including those with an ID that is equal to the integer "323126" or "323816". Depending

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/webapp" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <filters>
    <Criteria field="id" operator="IN">323126,323816</Criteria>
```

```
  </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>0</count>
</ServiceResponse>
```

Search web applications

Returns a list of web applications which are in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/search/was/webapp

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: WebApp for descriptions of all <WebApp> elements.

id (Integer)	createdDate (Date)
name (Text)	updatedDate (Date)
url (Text)	isScheduled (Boolean)
tags	isScanned (Boolean)
tags.name (Text)	lastScan.date (Date)
tags.id (Integer)	lastScan.status (Keyword: SUBMITTED, RUNNING, FINISHED, ERROR or CANCELLED)

Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN
Boolean	(true/false) EQUALS, NOT EQUALS

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Output includes web applications within the user's scope

Examples

Example 1: Search - no criteria (POST)

Return a list of all the web applications in the user's account.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml"
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/webapp" -X
"POST"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
  <hasMoreRecords>false</hasMoreRecords>
  <lastId>323103</lastId>
  <data>
    <WebApp>
      <id>323102</id>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[https://example.com]]></url>
      <owner>
        <id>123068</id>
      </owner>
      <tags>
        <count>3</count>
      </tags>
      <createdDate>2008-11-22T13:48:03Z</createdDate>
      <updatedDate>2011-12-19T13:41:07Z</updatedDate>
    </WebApp>
    <WebApp>
      <id>323103</id>
      <name><![CDATA[Demo Web App]]></name>

<url><![CDATA[http://10.10.26.200:80/phpBB/1.4.4_basic]]></url>
      <owner>
        <id>123071</id>
```

```
      </owner>
      <tags>
        <count>0</count>
      </tags>
      <createdDate>2008-11-22T13:45:46Z</createdDate>
      <updatedDate>2011-12-16T14:33:38Z</updatedDate>
    </WebApp>
  </data>
</ServiceResponse>
```

Example 2: Search - criteria (POST)

Return a list of web applications in the user's account that have a name containing the word "Merchant" and an ID greater than 323000.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/webapp" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
 <filters>
   <Criteria field="name" operator="CONTAINS">Merchant</Criteria>
   <Criteria field="id" operator="GREATER">323000</Criteria>
 </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <WebApp>
```

```
      <id>323476</id>
      <name><![CDATA[Merchant site 1]]></name>

<url><![CDATA[http://10.10.25.116:80/merchant/2.2/themerchant]]></
url>
      <owner>
        <id>123056</id>
      </owner>
      <tags>
        <count>0</count>
      </tags>
      <createdDate>2011-02-21T15:24:49Z</createdDate>
      <updatedDate>2012-01-03T16:53:37Z</updatedDate>
    </WebApp>
  </data>
</ServiceResponse>
```

Get details for a web application

Returns details for a web application which is in the user's scope. Want to find a web application ID to use as input? See Search web applications.

URL: https://qualysapi.qualys.com/qps/rest/3.0/get/was/webapp/<id>

Methods allowed: GET

The web application screenshot, when available, is included in the output in the "screenshot" element as a base64 encoded binary string. This string needs to be converted before a user can decode and view the image file (.png). In order to encode screenshots we use urlSafe Base 64 encoding solution like other elements in our API. Therefore these characters will be replaced in the base64 contents:

/ will be replaced with _

+ will be replaced with -

Input

The element "id" (Integer) is required, where "id" identifies a web application.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Web application must be within the user's scope

Example

Details - criteria (GET)

View details for the web application with the ID 95345.

Request:

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/webapp/2130421"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
```

```
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>2130421</id>
      <name><![CDATA[CUSTOM PARAM TEST]]></name>
      <url><![CDATA
[http://funkytown.acme01.acme.com/Forms/FormFields/temp/]]></url>
      <os>Linux 2.4-2.6 / Embedded Device / F5 Networks Big-IP /
Linux 2.6</os>
      <owner>
        <id>4354</id>
        <username>acme_as</username>
        <firstName><![CDATA[Alex]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
      <scope>ALL</scope>
      <attributes>
        <count>0</count>
      </attributes>
      <defaultProfile>
        <id>139359</id>
        <name><![CDATA[10 Links edit]]></name>
      </defaultProfile>
      <defaultScanner>
        <type>EXTERNAL</type>
      </defaultScanner>
      <scannerLocked>false</scannerLocked>
      <urlBlacklist>
        <count>0</count>
      </urlBlacklist>
      <urlWhitelist>
        <count>0</count>
      </urlWhitelist>
      <postDataBlacklist>
        <count>0</count>
      </postDataBlacklist>
      <authRecords>
        <count>1</count>
        <list>
          <WebAppAuthRecord>
```

```
      <id>127357</id>
      <name><![CDATA[AR - funkytown]]></name>
    </WebAppAuthRecord>
  </list>
</authRecords>
<useRobots>IGNORE</useRobots>
<useSitemap>false</useSitemap>
<malwareMonitoring>true</malwareMonitoring>
<malwareNotification>true</malwareNotification>
<malwareScheduleTime>18:00</malwareScheduleTime>
<malwareScheduleTimeZone>
  <code>America/Dawson</code>
  <offset>-07:00</offset>
</malwareScheduleTimeZone>
<tags>
  <count>4</count>
  <list>
    <Tag>
      <id>1730872</id>
      <name><![CDATA[new tag]]></name>
    </Tag>
    <Tag>
      <id>1418973</id>
      <name><![CDATA[Cert Tag]]></name>
    </Tag>
    <Tag>
      <id>1693034</id>
      <name><![CDATA[My Tag name]]></name>
    </Tag>
    <Tag>
      <id>1693032</id>
      <name><![CDATA[Groovy tag -1]]></name>
    </Tag>
  </list>
</tags>
<comments>
  <count>0</count>
</comments>
<isScheduled>false</isScheduled>
<lastScan>
  <id>827468</id>
```

```
        <name><![CDATA[Web Application Vulnerability Scan - CUSTOM
PARAM TEST]]></name>
      </lastScan>
      <createdBy>
        <id>4354</id>
        <username>acme_as</username>
        <firstName><![CDATA[Alex]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </createdBy>
      <createdDate>2014-07-24T09:08:49Z</createdDate>
      <updatedBy>
        <id>4354</id>
        <username>acme_as</username>
        <firstName><![CDATA[Alex]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
      <updatedDate>2014-09-24T23:34:17Z</updatedDate>

<screenshot><![CDATA[_9j_4AAQSkZJRgABAQEAegBrAAD_2wBDAAYEBQYFBAYGB
QYHBwYIChAKCgkJChQODwwQFxQYGBcUFhYaHSUfGhsjHBYWICwgIyYnKSopGR8tMC0
oMCUoKSj_2wBDAQcHBwoIChMKChMoGhYaKCgoKCgoKCgoKCgoKCgoKCgoKCgoK
CgoKCgoKCg
… (shortened for brevity)

UrlSafe encoded. Convert this value in order to decode and view the
image file (.png).

</screenshot>
    </WebApp>
  </data>
</ServiceResponse>
```

Create a web application

A web application is a configuration in your account. Once created, a user can select the web application as the target of a web application scan.

URL: https://qualysapi.qualys.com/qps/rest/3.0/create/was/webapp

Methods allowed: POST

Input

Required elements are “name” and “url”. Other elements are optional. All dates must be entered in UTC date/time format. See Reference: WebApp for descriptions of all <WebApp> elements.

When only “name” and “url” are specified:

- Scope defaults to ALL. The scanner will crawl all directories and sub-directories of the starting URL.
- No default option profile is specified. An option profile must be specified for each scan.
- No authentication records are defined. No form or server authentication will be performed.
- No blacklists or whitelists are defined. All directories and sub-directories of the starting URL will be scanned.

Permissions

User must have the WAS application enabled
User must have “API Access” permission
User must have “Create Web Asset” permission

Examples

Example 1: Create - minimum criteria (POST)

Create a new web application called "My Web Application" that has the starting URL "http://mywebapp.com". The default web application settings are assigned automatically.

Request:

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/webapp/" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST data:

```
<ServiceRequest>
  <data>
    <WebApp>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://mywebapp.com]]></url>
    </WebApp>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>1912949</id>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://mywebapp.com]]]></url>
      <owner>
        <id>45941</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
```

```
  <lastName><![CDATA[Smith]]></lastName>
</owner>
<scope>ALL</scope>
<attributes>
  <count>0</count>
</attributes>
<defaultScanner>
  <type>EXTERNAL</type>
</defaultScanner>
<scannerLocked>false</scannerLocked>
<urlBlacklist>
  <count>0</count>
</urlBlacklist>
<urlWhitelist>
  <count>0</count>
</urlWhitelist>
<postDataBlacklist>
  <count>0</count>
</postDataBlacklist>
<authRecords>
  <count>0</count>
</authRecords>
<useRobots>IGNORE</useRobots>
<useSitemap>false</useSitemap>
<malwareMonitoring>false</malwareMonitoring>
<tags>
  <count>0</count>
</tags>
<comments>
  <count>0</count>
</comments>
<isScheduled>false</isScheduled>
<createdBy>
  <id>45941</id>
  <username>username</username>
  <firstName><![CDATA[John]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</createdBy>
<createdDate>2013-10-18T18:26:40Z</createdDate>
<updatedBy>
  <id>45941</id>
```

```
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
      <updatedDate>2013-10-18T18:26:40Z</updatedDate>
    </WebApp>
  </data>
</ServiceResponse>
```

Example 2: Create with 1 authentication record (POST)

Create a new web application called "My Web Application" that has the starting URL "http://mywebapp.com" and has 1 authentication record.

Request:

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/webapp/" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST data:

```
<ServiceRequest>
  <data>
    <WebApp>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://mywebapp.com]]></url>
      <authRecords>
        <set>
          <WebAppAuthRecord>
            <id>77350</id>
          </WebAppAuthRecord>
        </set>
     </authRecords>
    </WebApp>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
```

```
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>1929030</id>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://mywebapp.com]]]></url>
      <owner>
        <id>45941</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
      <scope>ALL</scope>
      <attributes>
        <count>0</count>
      </attributes>
      <defaultScanner>
        <type>EXTERNAL</type>
      </defaultScanner>
      <scannerLocked>false</scannerLocked>
      <urlBlacklist>
        <count>0</count>
      </urlBlacklist>
      <urlWhitelist>
        <count>0</count>
      </urlWhitelist>
      <postDataBlacklist>
        <count>0</count>
      </postDataBlacklist>
      <authRecords>
        <count>1</count>
        <list>
          <WebAppAuthRecord>
            <id>77350</id>
            <name><![CDATA[My Authentication Record]]></name>
          </WebAppAuthRecord>
        </list>
      <useRobots>IGNORE</useRobots>
```

```
    ...
   </WebApp>
  </data>
</ServiceResponse>
```

Example 3: Create - all criteria (POST)

Create a new web application with the name "My Web Application" and the starting URL "http://www.example.com". The web application is assigned custom settings as defined in the request POST data.

Request:

```
curl -n "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/webapp/" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST data:

```
<ServiceRequest>
     <data>
          <WebApp>
               <name><![CDATA[My Web Application]]></name>
                    <url> <![CDATA[http://www.example.com]]></url>
                        <scope>DOMAINS</scope>
                        <domains>
<set>
<Domain><![CDATA[corp2.ab.myapp.com]]></Domain>
<Domain><![CDATA[corp1.myapp.com]]></Domain>

</set>
</domains>
     <uris>
          <set>
<Url><![CDATA[http://corp1.myapp.com]]></Url>
<Url><![CDATA[http://corp1.myapp.com/]]></Url>
<Url><![CDATA[https://corp1.myapp.com]]></Url>
<Url><![CDATA[https://corp1.myapp.com/]]></Url>
<Url><![CDATA[https://corp1.myapp.com:443]]></Url>
<Url><![CDATA[https://corp1.myapp.com:443/]]></Url>
<Url><![CDATA[http://corp1.myapp.com:8080/]]></Url>
<Url><![CDATA[http://corp1.myapp.com/startingUri]]></Url>
```

```
<Url><![CDATA[http://corp1.myapp.com/startingUri?]]></Url>
<Url><![CDATA[http://corp1.myapp.com/startingUri?param=true]]>
</Url>
<Url><![CDATA[http://corp1.myapp.com/startingUri?param=true&param2
=false]]></Url>
<Url><![CDATA[http://corp1.myapp.com/otherUri]]></Url>
<Url><![CDATA[http://corp1.myapp.com/otherUri?param=1]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com/]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com/]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com:443]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com:443/]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com:8080/]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com/startingUri]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com/startingUri?]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com/startingUri?param=true]]><
/Url>
<Url><![CDATA[http://corp2.ab.myapp.com:443/startingUri?param=true
&param2=false]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com:8080/otherUri]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com/otherUri?param=1]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com/otherUri?param=1]]></Url>

</set>
</uris>
     <attributes>
          <set>
               <Attribute>
                    <category>Business Function</category>
                         value><![CDATA[some business
function]]></value>
               </Attribute>
               <Attribute>
                    <category>Business Location</category>
                         <value><![CDATA[some business
location]]></value>
               </Attribute>
               <Attribute>
                    <category>Business Description</category>
                         <value><![CDATA[some business
description]]></value>
```

```
                </Attribute>
            </set>
        </attributes>
        <tags><set>
        <Tag><id>217118</id></Tag>
        <Tag><id>152743</id></Tag>
        <Tag><id>216368</id></Tag>
        <Tag><id>153442</id></Tag>
        </set>
        </tags>
        <defaultProfile>
            <id>90212</id>
        </defaultProfile>
        <defaultScanner>
            <type>INTERNAL</type>
            <friendlyName><![CDATA[friendlyname]]>
</friendlyName>
        </defaultScanner>
        <useRobots>BLACKLIST</useRobots>
        <useSitemap>true</useSitemap>
        <headers>
            <set>
                <WebAppHeader><![CDATA[some headers]]>
</WebAppHeader>
            </set>
        </headers>
        <urlBlacklist>
            <set>
                <UrlEntry regex="true">
<![CDATA[http://rg.blacklist.*.qa.myapp.com]]></UrlEntry>
                <UrlEntry regex="true">
<![CDATA[http://rg.blacklist.*?]]></UrlEntry>
                <UrlEntry>
<![CDATA[http://url.blacklist.2.ab.myapp.com]]></UrlEntry>
                <UrlEntry regex="false">
<![CDATA[http://url.blacklist.3.qa.myapp.com]]></UrlEntry>
            </set>
        </urlBlacklist>
        <urlWhitelist>
            <set>
                <UrlEntry regex="true">
<![CDATA[http://rg.whitelist.*.qa.myapp.com]]></UrlEntry>
```

```
                    <UrlEntry regex="true">
<![CDATA[http://rg.whitelist.*?]]></UrlEntry>

<UrlEntry><![CDATA[http://url.whitelist.2.ab.myapp.com]]></UrlEntr
y><UrlEntry regex="false"><![CDATA

[http://url.whitelist.3.ab.myapp.com]]></UrlEntry>
               </set>
          </urlWhitelist>
          <postDataBlacklist>
               <set>
                    <UrlEntry regex="true"><![CDATA
[http://rg.postdatblacklist.*.ab.myapp.com]]></UrlEntry>
                    <UrlEntry
regex="true"><![CDATA[http://rg.postdatblacklist.*?]]></UrlEntry>
               </set>
          </postDataBlacklist>
          <comments>
               <set>
                    <Comment>
                    <contents><![CDATA[some additional
comments]]></contents>
                    </Comment>
               </set>
          </comments>
          </WebApp>
     </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>1912750</id>
      <name><![CDATA[My Web Application]]></name>
```

```
        <url><![CDATA[http://www.example.com]]></url>
        <owner>
          <id>45941</id>
          <username>username</username>
          <firstName><![CDATA[John]]></firstName>
          <lastName><![CDATA[Smith]]></lastName>
        </owner>
        <scope>DOMAINS</scope>
        <domains>
          <count>2</count>
          <list>
            <Domain><![CDATA[corp1.myapp.com]]></Domain>
            <Domain><![CDATA[corp2.ab.myapp.com]]></Domain>
          </list>
        </domains>
        <uris>
          <count>26</count>
          <list>
            <Url><![CDATA[https://corp2.ab.myapp.com]]></Url>

<Url><![CDATA[http://corp1.myapp.com/otherUri?param=1]]></Url>
            <Url><![CDATA[http://corp1.myapp.com/]]></Url>
            <Url><![CDATA[https://corp1.myapp.com]]></Url>

<Url><![CDATA[http://corp1.myapp.com/startingUri?]]></Url>
            <Url><![CDATA[https://corp2.ab.myapp.com:443/]]></Url>

<Url><![CDATA[https://corp2.ab.myapp.com/otherUri?param=1]]></Url>
            <Url><![CDATA[https://corp1.myapp.com:443/]]></Url>

<Url><![CDATA[http://corp2.ab.myapp.com/startingUri?param=true]]><
/Url>
            <Url><![CDATA[http://corp2.ab.myapp.com:8080/]]></Url>
            <Url><![CDATA[http://corp1.myapp.com/otherUri]]></Url>

<Url><![CDATA[http://corp1.myapp.com/startingUri?param=true&param2
=false]]></Url>
            <Url><![CDATA[http://corp1.myapp.com]]></Url>

<Url><![CDATA[http://corp1.myapp.com/startingUri?param=true]]></Ur
l>
            <Url><![CDATA[http://corp2.ab.myapp.com]]></Url>
```

```
          <Url><![CDATA[https://corp2.ab.myapp.com/]]></Url>
          <Url><![CDATA[http://corp2.ab.myapp.com/]]></Url>
          <Url><![CDATA[https://corp2.ab.myapp.com:443]]></Url>
         <Url><![CDATA[http://corp1.myapp.com/startingUri]]></Url>
          <Url><![CDATA[https://corp1.myapp.com:443]]></Url>

<Url><![CDATA[http://corp2.ab.myapp.com/startingUri]]></Url>
          <Url><![CDATA[http://corp1.myapp.com:8080/]]></Url>

<Url><![CDATA[https://corp2.ab.myapp.com:8080/otherUri]]></Url>
          <Url><![CDATA[https://corp1.myapp.com/]]></Url>

<Url><![CDATA[http://corp2.ab.myapp.com/startingUri?]]></Url>

<Url><![CDATA[http://corp2.ab.myapp.com:443/startingUri?param=true
&param2=false]]></Url>
        </list>
      </uris>
      <attributes>
        <count>3</count>
        <list>
          <Attribute>
            <category>Business Function</category>
            <value><![CDATA[some business function]]></value>
          </Attribute>
          <Attribute>
            <category>Business Description</category>
            <value><![CDATA[some business description]]></value>
          </Attribute>
          <Attribute>
            <category>Business Location</category>
            <value><![CDATA[some business location]]></value>
          </Attribute>
        </list>
      </attributes>
      <defaultProfile>
        <id>90212</id>
        <name><![CDATA[Initial WAS Options]]></name>
      </defaultProfile>
      <defaultScanner>
        <type>INTERNAL</type>
        <friendlyName><![CDATA[friendlyname]]></friendlyName>
```

```
      </defaultScanner>
      <scannerLocked>false</scannerLocked>
      <urlBlacklist>
        <count>4</count>
        <list>
          <UrlEntry
regex="false"><![CDATA[http://url.blacklist.2.ab.myapp.com]]></Url
Entry>
          <UrlEntry
regex="false"><![CDATA[http://url.blacklist.3.ab.myapp.com]]></Url
Entry>
          <UrlEntry
regex="true"><![CDATA[http://rg.blacklist.*.ab.myapp.com]]></UrlEn
try>
          <UrlEntry
regex="true"><![CDATA[http://rg.blacklist.*?]]></UrlEntry>
        </list>
      </urlBlacklist>
      <urlWhitelist>
        <count>4</count>
        <list>
          <UrlEntry
regex="true"><![CDATA[http://rg.whitelist.*.ab.myapp.com]]></UrlEn
try>
          <UrlEntry
regex="true"><![CDATA[http://rg.whitelist.*?]]></UrlEntry>
          <UrlEntry
regex="false"><![CDATA[http://url.whitelist.2.ab.myapp.com]]></Url
Entry>
          <UrlEntry
regex="false"><![CDATA[http://url.whitelist.3.ab.myapp.com]]></Url
Entry>
        </list>
      </urlWhitelist>
      <postDataBlacklist>
        <count>2</count>
        <list>
          <UrlEntry
regex="true"><![CDATA[http://rg.postdatblacklist.*.ab.myapp.com]]>
</UrlEntry>
          <UrlEntry
regex="true"><![CDATA[http://rg.postdatblacklist.*?]]></UrlEntry>
        </list>
```

```
      </postDataBlacklist>
      <authRecords>
        <count>0</count>
      </authRecords>
      <useRobots>BLACKLIST</useRobots>
      <useSitemap>true</useSitemap>
      <headers>
        <count>1</count>
        <list>
          <WebAppHeader><![CDATA[some headers]]></WebAppHeader>
        </list>
      </headers>
      <malwareMonitoring>false</malwareMonitoring>
      <tags>
        <count>4</count>
        <list>
          <Tag>
            <id>152743</id>
            <name><![CDATA[Asset Groups]]></name>
          </Tag>
          <Tag>
            <id>217118</id>
            <name><![CDATA[AUG 27]]></name>
          </Tag>
          <Tag>
            <id>153442</id>
            <name><![CDATA[Malware Domain Assets]]></name>
          </Tag>
          <Tag>
            <id>216368</id>
            <name><![CDATA[Asset name rule]]></name>
          </Tag>
        </list>
      </tags>
      <comments>
        <count>1</count>
        <list>
          <Comment>
            <contents><![CDATA[some additional
comments]]></contents>
            <createdDate>2013-10-18T17:57:32Z</createdDate>
```

```
            </Comment>
          </list>
        </comments>
        <isScheduled>false</isScheduled>
        <createdBy>
          <id>45941</id>
          <username>username</username>
          <firstName><![CDATA[John]]></firstName>
          <lastName><![CDATA[Smith]]></lastName>
        </createdBy>
        <createdDate>2013-10-18T17:57:32Z</createdDate>
        <updatedBy>
          <id>45941</id>
          <username>username</username>
          <firstName><![CDATA[John]]></firstName>
          <lastName><![CDATA[Smith]]></lastName>
        </updatedBy>
        <updatedDate>2013-10-18T17:57:32Z</updatedDate>
      </WebApp>
    </data>
  </ServiceResponse>
```

Update a web application

Update a web application configuration in your account.

URL: https://qualysapi.qualys.com/qps/rest/3.0/update/was/webapp/<id>

Methods allowed: POST

Input

The "id" (Integer) element is required where "id" identifies a web application. Additional elements are optional. See Reference: WebApp for descriptions of all <WebApp> elements.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Edit Web Asset" permission
Web application must be within the user's scope

Examples

Example 1: Update - minimum information (POST)

For the web application with ID 1234, change the name to "My WebApp Name" .

Request:

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/webapp/1234"
< file.xml
```

Note: "file.xml" contains the request POST data.

Request POST data (file.xml):

```
<ServiceRequest>
  <data>
    <WebApp>
      <name>My WebApp Name</name>
    </WebApp>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>1234</id>
    </WebApp>
  </data>
</ServiceResponse>
```

Example 2: Update - authentication records (POST)

For the web application with ID 1234, add 1 authentication record and remove 1 authentication record..

Request:

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/webapp/1234"
< file.xml
```

Note: “file.xml” contains the request POST data.

Request POST data (file.xml):

```
<ServiceRequest>
  <data>
    <WebApp>
      <name><![CDATA[My WebApp Name]]></name>
      <authRecords>
        <add>
          <WebAppAuthRecord>
            <id>77355</id>
          </WebAppAuthRecord>
        </add>
        <remove>
          <WebAppAuthRecord>
            <id>77356</id>
          </WebAppAuthRecord>
```

```
        </remove>
      </authRecords>
    </WebApp>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>1234</id>
    </WebApp>
  </data>
</ServiceResponse>
```

Example 2: Update - multiple settings (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/webapp/32347
7"
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
    <WebApp>
      <name>My Web Application</name>
      <url>http://mywebapp.com</url>
      <scope>DOMAINS</scope>
      <domains>
        <remove>
          <Domain>abc.com</Domain>
```

```
            <Domain>def.com</Domain>
            <Domain>ghi.com</Domain>
          </remove>
        </domains>
        <attributes>
          <remove>
            <Attribute>
              <category>Business Function</category>
             </Attribute>
            <Attribute>
              <category>Business Location</category>
            </Attribute>
          </remove>
          <update>
            <Attribute>
              <category>Business Description</category>
              <value>Business Description Value - UPDATED</value>
            </Attribute>
          </update>
        </attributes>
        <defaultProfile><id>1024</id></defaultProfile>
        <defaultScanner>
          <type>INTERNAL</type>
          <friendlyName>SA 123</friendlyName>
        </defaultScanner>
        <urlBlacklist>
          <set>
<UrlEntry><![CDATA[http://url.blacklist.1.mywebapp.com]]></UrlEntr
y>
            <UrlEntry
regex="false"><![CDATA[http://url.blacklist.2.mywebapp.com]]></Url
Entry>
            <UrlEntry
regex="true"><![CDATA[http://rg.blacklist.*.com]]></UrlEntry>
          </set>
        </urlBlacklist>
        <urlWhitelist>
          <set>
<UrlEntry><![CDATA[http://url.whitelist.1.mywebapp.com]]></UrlEntr
y>
          <UrlEntry
regex="false"><![CDATA[http://url.whitelist.2.mywebapp.com]]></Url
```

```
Entry>
        <UrlEntry
regex="true"><![CDATA[http://rg.whitelist.*.mywebapp.com]]></UrlEn
try>
        </set>
      </urlWhitelist>
      <postDataBlacklist>
        <set>
        <UrlEntry
regex="true"><![CDATA[http://url.postdatablacklist.1.mywebapp.com]
]></UrlEntry>
        <UrlEntry
regex="true"><![CDATA[http://url.postdatablacklist.2.mywebapp.com]
]></UrlEntry>
        </set>
      </postDataBlacklist>
      <useRobots>ADD_PATHS</useRobots>
      <useSitemap>true</useSitemap>
      <headers>
        <set>
          <WebAppHeader>X-TTP-REQUESTED-BY: Qualys
Test</WebAppHeader>
        </set>
      </headers>
    </WebApp>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>324377</id>
    </WebApp>
  </data>
</ServiceResponse>
```

Example 3: Update - set default cancel time (POST)

Set the default cancel scan option for web application ID 2392272. Scans of this web application will be set to cancel at 10pm by default.

Request:

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/webapp/23922
72" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST data (file.xml):

```
<ServiceRequest>
  <data>
  <WebApp>
    <name><![CDATA[My Web App]]></name>
    <url><![CDATA[http://mywebapp.com]]></url>
    <config><cancelScansAt>22:00</cancelScansAt></config>
  </WebApp>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.qualys.com
/qps/xsd/3.0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>2392272</id>
    </WebApp>
  </data>
</ServiceResponse>
```

Delete web applications

Delete a web application configuration in your account.

URL: https://qualysapi.qualys.com/qps/rest/3.0/delete/was/webapp/<id>
https://qualysapi.qualys.com/qps/rest/3.0/delete/was/webapp/<filters>

Methods allowed: POST

Input

The "id" (Integer) element is required where "id" identifies a web application. Additional elements are optional. See Reference: WebApp for descriptions of all <WebApp> elements.

id (Integer)	createdDate (Date)
name (Text)	updatedDate (Date)
url (Text)	isScheduled (Boolean)
tags.name (Text)	isScanned (Boolean)
tags.id (Integer)	lastScan.status (Keyword: SUBMITTED, RUNNING, FINISHED, ERROR or CANCELLED)
	lastScan.date (Date)

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Delete Web Asset" permission
Web application must be within the user's scope

Examples

Example 1: Delete - single (POST)

Delete the web application that has the ID 1234.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/webapp/1234"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
   <count>1</count>
    <data>
      <WebApp>
        <id>1234</id>
      </WebApp>
    </data>
</ServiceResponse>
```

Example 2: Delete bulk (POST)

Delete web applications in the user's account that have a name with the word "Merchant" and have an ID greater than 323000.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
https://qualysapi.qualys.com/qps/rest/3.0/delete/was/webapp/ <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST data:

```
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">Merchant</Criteria>
    <Criteria field="id" operator="GREATER">323000</Criteria>
  </filters>
</ServiceRequest>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
```

```
  <count>3</count>
  <data>
    <WebApp>
      <id>323126</id>
    </WebApp>
    <WebApp>
      <id>324256</id>
    </WebApp>
    <WebApp>
      <id>323476</id>
    </WebApp>
  </data>
</ServiceResponse>
```

Purge web applications

All dates must be entered in UTC date/time format. Purges a web application which is in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/purge/was/webapp/<id>
or
https://qualysapi.qualys.com/qps/rest/3.0/purge/was/webapp/<filters>

Methods allowed: POST

Input

The "id" (Integer) element is required where "id" identifies a web application. Additional elements are optional. See Reference: WebApp for descriptions of all <WebApp> elements.

id (Integer)	createdDate (Date)
name (Text)	updatedDate (Date)
url (Text)	isScheduled (Boolean)
tags.name (Text)	isScanned (Boolean)
tags.id (Integer)	lastScan.status (Keyword: SUBMITTED, RUNNING, FINISHED, ERROR or CANCELLED)
	lastScan.date (Date)

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Purge Web Asset" permission
Web application must be within the user's scope

Examples

Example 1: Purge single (POST)

Purge the web application that has the ID 32420.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml"
"https://qualysapi.qualys.com/qps/rest/3.0/purge/was/webapp/32420"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
 <responseCode>SUCCESS</responseCode>
 <count>1</count>
  <data>
    <WebApp>
      <id>32420</id>
    </WebApp>
  </data>
</ServiceResponse>
```

Example 2: Purge bulk (POST)

Purge web applications in the user's account that have a name with the word "Merchant" and have an ID greater than 323000.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
https://qualysapi.qualys.com/qps/rest/3.0/purge/was/webapp/ <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST data:

```
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">Merchant</Criteria>
    <Criteria field="id" operator="GREATER">323000</Criteria>
```

```
  </filters>
</ServiceRequest>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>3</count>
  <data>
    <WebApp>
      <id>323126</id>
    </WebApp>
    <WebApp>
      <id>324256</id>
    </WebApp>
    <WebApp>
      <id>323476</id>
    </WebApp>
  </data>
</ServiceResponse>
```

Reference: WebApp

The <WebApp> element includes sub elements used to define a web application. A reference of these elements is provided below. An asterisk * indicates a complex element.

Element (data type)	Description
Web Application	
name (Text)	The web application name (maximum 256 characters). This element is required to create a web application.
url (Text)	The URL of the web application (maximum 2048 characters). This element is required to create a web application.
id (Integer)	The web application ID. This element is assigned by the service and required for an update request.
os	The operating system of the web application.
owner (Text)	This element is assigned by the service and may be specified for an update request only.

config*	Configure the cancel scan option. Specify "cancel after" time or "cancel at" time. Only one of `<cancelScansAfterNHours>` or `<cancelScanstAt>` is allowed in one config section.
	
	Example for "cancel after" time:
	`<config>`
	`  <cancelScansAfterNHours>3`
	`</cancelScansAfterNHours>`
	`</config>`
	
	Example for "cancel at" time:
	`<config>`
	`  <cancelScansAt>2015-06-10T12:00:00Z`
	`</cancelScansAt>`
	`</config>`
	
	Notes about updating web applications
	- If none of the above elements are specified in the config section, the default cancel option is removed from the web app settings.
	- If the config section is not specified, not changes are made to the web app settings.

Element (data type)	Description

attributes*	Web application attributes set as categories: Business Function, Business Description and Business Location.
	Example:
	`<attributes>`
	`<set>`
	`<Attribute>`
	`<category>Business Function</category>`
	`<value><![CDATA[function]]></value>`
	`</Attribute>`
	`<Attribute>`
	`<category>Business Location</category>`
	`<value><![CDATA[location]]></value>`
	`</Attribute>`
	`<Attribute>`
	`<category>Business Description</category>`
	`<value><![CDATA[description]]></value>`
	`</Attribute>`
	`</set>`
	`</attributes>`

tags*	Tags assigned to the web application.
	Example:
	`<tags>`
	`<set>`
	`<Tag>`
	`<id>12345</id>`
	`</Tag>`
	`<Tag>`
	`<id>12345678</id>`
	`</Tag>`
	`</set>`
	`</tags>`

comments (Text)	Comments on the web application.
scope (Keyword)	The scanning scope for the web application: ALL (default), LIMIT, SUBDOMAIN or DOMAINS. If set to ALL, the scan will crawl all directories and sub-directories of the starting URL. If set to LIMIT, crawling will be limited to the starting URI's initial path and sub-directories. If set to SUBDOMAINS, any sub-domain that is in the same domain as the specified domain name will be crawled. If set to DOMAINS, only the specified domains will be crawled.
uris (Text)	Additional URLs to crawl. Each must be a valid HTTP or HTTPS URL consistent with the web application scope.

Element (data type)	Description
malwareMonitoring (Boolean)	A flag indicating whether Malware Monitoring is enabled for the web application (Boolean). Example: `<malwareMonitoring>true</malwareMonitoring>`
malwareNotification (Boolean)	A flag indicating whether email notification is enabled for Malware Monitoring scans (Boolean). Example: `<malwareNotification>true</malwareNotification>`
malwareScheduleTime	The scheculed time for the daily Malware Monitoring scan. Example: `<malwareScheduleTime>02:00</malwareScheduleTime>`

malwareScheduleTimeZone	The time zone that applies to the schedule for the Malware Monitoring Scan.
	Example:
	`<malwareScheduleTimeZone>`
	`  <code>America/Vancouver</code>`
	`</malwareScheduleTimeZone>`

Scan Settings	

defaultProfile*	The default option profile for scanning the web application. When unspecified, an option profile must be specified by the user for each scan.
	`<defaultProfile>`
	`  <id>139359</id>`
	`  <name><![CDATA[10 Links edit]]></name>`
	`</defaultProfile>`

defaultScanner*	The default scanner for the web application. A default scanner is optional. For type (Keyword) specify INTERNAL for a scanner appliance, or EXTERNAL for the external scanners. If type is INTERNAL, specify friendlyName (Text).
	Example:
	`<defaultScanner>`
	`  <type>INTERNAL</type>`
	`  <friendlyName>dp_scanner</friendlyName>`
	`</defaultScanner>`

proxy.id (Integer)	The default proxy for scanning the web application.
	Example:
	`<proxy>`
	`  <id>12345</id>`
	`</proxy>`

scannerLocked (Boolean)	A flag indicating whether the default scanner appliance is locked for the web application (Boolean). Example: `<scannerLocked>false</scannerLocked>`

Element (data type)	Description
useRobots (Keyword)	A flag indicating whether to observe the Robots.txt file and its directives if found when scanning the web application. If set to IGNORE (default) the Robots.txt file is ignANDed. If set to ADD_PATHS, the "disallow" and "allow" directives in the Robots.txt file will be observed; this means these directives will be added as link hints for the crawler. If set to BLACKLIST the "disallow" directives in the Robots.txt file will be observed; this means scans will not crawl matching links.
useSitemap (Boolean)	A flag indicating whether to adhere to a sitemap.xml file if present in the web application: true or false (default).

headers*	The headers that need to be injected by the scanning engine to scan the web application for complex authentication schemes or to impersonate a web browser.
	Example:
	`<headers>`
	`<set>`
	`<WebAppHeader>X-TTP-REQUESTED-BY: Qualys Test1</WebAppHeader>`
	`<WebAppHeader>X-TTP-REQUESTED-BY: Qualys Test2</WebAppHeader>`
	`</set>`
	`</headers>`

urlBlacklist*	The URLs for the black list. These are web application links (URLs) that you do not want scanned. For each URL, specify UrlEntry (Text). If the attribute regex (Boolean) is set to "true" the service performs a regular expression match.
	Example:
	`<urlBlacklist>`
	`<set>`
	`<UrlEntry><![CDATA[http://url.blacklist.1.xxx.com]]></UrlEntry>`
	`<UrlEntry regex="false"><![CDATA[http://url.blacklist.2.xxx.com]]></UrlEntry>`
	`<UrlEntry regex="true"><![CDATA[http://rg.blacklist.*.xxx.com]]></UrlEntry>`
	`</set>`
	`</urlBlacklist>`

Element (data type)	Description

urlWhitelist*	The URLs for the white list. These are web application links (URLs) that you want to be scanned. For each URL, specify UrlEntry (Text). If the attribute regex (Boolean) is set to "true" the service performs a regular expression match.
	Example:
	`<urlWhitelist>`
	`<set>`
	`<UrlEntry><![CDATA[http://url.whitelist.1.xxx.com]]></UrlEntry>`
	`<UrlEntry regex="false"><![CDATA[http://url.whitelist.2.xxx.com]]></UrlEntry>`
	`<UrlEntry regex="true"><![CDATA[http://rg.whitelist.*.xxx.com]]></UrlEntry>`
	`</set>`
	`</urlWhitelist>`

postDataBlacklist*	The web application URLs for which you want to block form submission (POST data), as this could have unwanted side effects. For each URL, specify UrlEntry (Text). The attribute regex (Boolean) can be set to “true” for a regular expression match.
	Example:
	`<postDataBlacklist>`
	`<set>`
	`<UrlEntry regex="true"><![CDATA[http://rg.postdatablacklist.*.qa.qualys.com]]></UrlEntry>`
	`<UrlEntry regex="true"><![CDATA[http://rg.postdatablacklist.*?]]></UrlEntry>`
	`</set>`
	`</postDataBlacklist>`

Element (data type)	Description

authRecords*	The web application authentication records. The WebAppAuthRecords element identifies a set of authentication instances (combination of form and types).
	Example:
	`<authRecords>`
	`<list>`
	`</list>`
	`<set>`
	`<WebAppAuthRecord>`
	`<name>Name of Record</name>`
	`<comments>Comments</comments>`
	`<formRecord>`
	`<type>STANDARD</type>`
	`<fields>`
	`...`
	`</authRecords>`

WebAppAuthRecord*	Under <authRecords>, this element identifies an authentication record assigned to the web application. Prior to WAS 3.1, authentication records and their settings were defined here using the Web Applicatin API. Now you can manage authentication records using the Authentication API.
Elements Assigned by the Service	
id (Integer)	The web application ID.
owner (Text)	The user login ID of the web application owner.
isScheduled (Boolean)	Is a scan scheduled for the web application? (true or false).
createdBy (Text)	The user who created the web application.
createdeDate (Date)	the date when the web application was created in UTC date/time format (2011-11-07T10:58:17Z).
updatedBy (Text)	The user who last updated the web application.
updatedDate (Date)	The date of the last update of the web application in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).
lastScan (Text)	The scan ID of the last scan run on the web application.
lastScan.status (Keyword)	The status of the most recent scan: SUBMITTED, RUNNING, FINISHED, ERROR or CANCELLED.

CHAPTER 3

Authentication API

Some web applications require authenticated access to most of their functionality. You can configure authentication by creating authentication records. Each record can be defined with multiple types - form authentication like login pages and server authentication (HTTP Basic, Digest, NTLM, SSL client certificates).

The WAS Authentication API provides a suite of API functions for managing authentication records. You can select a record as part of the web application settings thus making it the default (see the Web Application API), and you can select a record at scan time (see the Scan API and Schedule API).

These operations are available:

Current authentication record count

Search authentication records

Get details for an authentication record

Create a new authentication record

Update an authentication record

Delete authentication records

Current authentication record count

Returns the total number of authentication records in the user's scope. Input elements are optional and are used to filter the number of authentication records included in the count.

URL: https://qualysapi.qualys.com/qps/rest/3.0/count/was/webappauthrecord

Methods allowed: GET, POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

id (Integer)	updatedDate (Date)
name (Text)	lastScan.date (Date)
tags (Integer)	lastScan.authStatus (Keyword: NOT_USED, SUCCESSFUL, FAILED or PARTIAL)
tags.id (Integer)	isUsed (Boolean)
tags.name (Text)	contents (Keyword: FORM_STANDARD, FORM_CUSTOM, FORM_SELENIUM, SERVER_BASIC, SERVER_DIGEST)
createdDate (Date)	

Allowed Operators	
Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN
Boolean	(true/false) EQUALS, NOT EQUALS

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Count includes authentication records within the user's scope

Examples

Example 1: Count - no criteria (GET)

Return the number (count) of all authentication records in the user's scope.

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/webappauthrec
ord/"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>3</count>
</ServiceResponse>
```

Example 2: Count - criteria (POST)

Return the number (count) authentication records that have a name that contains the term "server".

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/webappauthrec
ord/" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">server</Criteria>
  </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
</ServiceResponse>
```

Search authentication records

Returns a list of authentication records which are in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/search/was/webappauthrecord

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

id (Integer)	updatedDate (Date)
name (Text)	lastScan.date (Date)
tags (Integer)	lastScan.authStatus (Keyword: NOT_USED, SUCCESSFUL, FAILED or PARTIAL)
tags.id (Integer)	isUsed (Boolean)
tags.name (Text)	contents (Keyword: FORM_STANDARD, FORM_CUSTOM, FORM_SELENIUM, SERVER_BASIC, SERVER_DIGEST)
createdDate (Date)	

Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN
Boolean	(true/false) EQUALS, NOT EQUALS

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Output includes authentication records within the user's scope

Examples

Example: Search - no criteria (POST)

Return a list of all authentication records in the user's scope.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/webappauthre
cord/"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.cm/qps/xsd
/3.0/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>3</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <WebAppAuthRecord>
      <id>82605</id>
      <name><![CDATA[Form Only]]></name>
      <owner>
        <id>630926</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
      <tags>
        <count>3</count>
      </tags>
      <createdDate>2013-10-24T04:32:14Z</createdDate>
      <updatedDate>2013-10-24T07:45:05Z</updatedDate>
    </WebAppAuthRecord>
    <WebAppAuthRecord>
      <id>82606</id>
      ...
    </WebAppAuthRecord>
    <WebAppAuthRecord>
      <id>82607</id>
```

```
      ...
    </WebAppAuthRecord>
  </data>
</ServiceResponse>
```

Example 2: Search - criteria (POST)

Return authentication record ID 82605.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/webappauthre
cord/" < file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <filters>
    <Criteria field="id" operator="EQUALS">82605</Criteria>
  </filters>
</ServiceRequest>
```

Get details for an authentication record

View details for an authentication record which is in the user's scope. Want to find a record ID to use as input? See Search authentication records.

URL: https://qualysapi.qualys.com/qps/rest/3.0/get/was/webappauthrecord/<id>

Methods allowed: GET

Input

The element "id" (Integer) is required, where "id" identifies an authentication record.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Authentication record must be within the user's scope

Example

View details (GET)

View details for authentication record ID 74078.

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/webappauthrecord/74078"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/was/webappauthrecord.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebAppAuthRecord>
      <id>74078</id>
      <name><![CDATA[My Authentication Record]]></name>
```

```
<owner>
  <id>4354</id>
  <username>username</username>
  <firstName><![CDATA[John]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</owner>
<formRecord>
  <type>STANDARD</type>
  <sslOnly>true</sslOnly>
  <fields>
    <count>2</count>
    <list>
      <WebAppAuthFormRecordField>
        <id>826453</id>
        <name><![CDATA[name1]]></name>
        <value><![CDATA[value]]></value>
      </WebAppAuthFormRecordField>
      <WebAppAuthFormRecordField>
        <id>826452</id>
        <name><![CDATA[name2]]></name>
        <value><![CDATA[value]]></value>
      </WebAppAuthFormRecordField>
    </list>
  </fields>
</formRecord>
<tags>
  <count>1</count>
  <list>
    <Tag>
      <id>1418973</id>
      <name><![CDATA[Cert Tag]]></name>
    </Tag>
  </list>
</tags>
<comments>
  <count>0</count>
</comments>
<createdDate>2013-09-23T20:21:04Z</createdDate>
<createdBy>
  <id>4354</id>
  <username>username</username>
```

```
          <firstName><![CDATA[John]]></firstName>
          <lastName><![CDATA[Smith]]></lastName>
        </createdBy>
        <updatedDate>2013-10-22T05:48:57Z</updatedDate>
        <updatedBy>
          <id>4354</id>
          <username>username</username>
          <firstName><![CDATA[John]]></firstName>
          <lastName><![CDATA[Smith]]></lastName>
        </updatedBy>
      </WebAppAuthRecord>
    </data>
  </ServiceResponse>
```

Create a new authentication record

Create a new authentication record.

URL: https://qualysapi.qualys.com/qps/rest/3.0/create/was/webappauthrecord

Methods allowed: POST

Input

The required input elements are listed below. The associated data type for each element appears in parentheses.

Required Elements	Optional Elements
name (Text)	tags
WebAuthRecord (Text)	comments

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Create Authentication Record" permission

Example

Example 1: Create - standard authentication (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/webappauthre
cord/" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
  <WebAppAuthRecord>
    <name><![CDATA[STANDARD auth]]></name>
    <formRecord>
      <type>STANDARD</type>
```

```
      <sslOnly>true</sslOnly>
      <fields>
        <set>
          <WebAppAuthFormRecordField>
            <name>username</name>
            <value>Login</value>
          </WebAppAuthFormRecordField>
          <WebAppAuthFormRecordField>
            <name>password</name>
            <value>Login with password</value>
          </WebAppAuthFormRecordField>
        </set>
      </fields>
    </formRecord>
    <tags>
      <set>
        <Tag>
          <id>152743</id>
        </Tag>
      </set>
    </tags>
    <comments>
      <set>
        <Comment><contents><![CDATA[some
comments]]></contents></Comment>
      </set>
    </comments>
  </WebAppAuthRecord>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebAppAuthRecord>
```

```
<id>80149</id>
<name><![CDATA[STANDARD auth]]></name>
<owner>
  <id>45941</id>
  <username>username</username>
  <firstName><![CDATA[John]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</owner>
<formRecord>
  <type>STANDARD</type>
  <sslOnly>true</sslOnly>
  <fields>
    <count>2</count>
    <list>
      <WebAppAuthFormRecordField>
        <id>835050</id>
        <name><![CDATA[username]]></name>
        <value><![CDATA[Login]]></value>
      </WebAppAuthFormRecordField>
      <WebAppAuthFormRecordField>
        <id>835051</id>
        <name><![CDATA[password]]></name>
        <value><![CDATA[*****]]></value>
      </WebAppAuthFormRecordField>
    </list>
  </fields>
</formRecord>
<tags>
  <count>1</count>
  <list>
    <Tag>
      <id>152743</id>
      <name><![CDATA[Asset Groups]]></name>
    </Tag>
  </list>
</tags>
<comments>
  <count>1</count>
  <list>
    <Comment>
      <contents><![CDATA[some comments]]></contents>
```

```
            <createdDate>2013-10-18T18:18:01Z</createdDate>
          </Comment>
        </list>
      </comments>
      <createdDate>2013-10-18T18:18:01Z</createdDate>
      <createdBy>
        <id>45941</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </createdBy>
      <updatedDate>2013-10-18T18:18:01Z</updatedDate>
      <updatedBy>
        <id>45941</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
    </WebAppAuthRecord>
  </data>
</ServiceResponse>
```

Example 2: Create - custom authentication (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/webappauthre
cord/" < file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
  <WebAppAuthRecord>
    <name><![CDATA[CUSTOM auth]]></name>
      <formRecord>
        <type>CUSTOM</type>
        <sslOnly>true</sslOnly>
        <fields>
```

```
          <set>
            <WebAppAuthFormRecordField>
              <name>some username</name>
              <value>Login</value>
              <secured>false</secured>
            </WebAppAuthFormRecordField>
            <WebAppAuthFormRecordField>
              <name>some password with true</name>
              <value>real password</value>
              <secured>true</secured>
            </WebAppAuthFormRecordField>
            <WebAppAuthFormRecordField>
              <name>not password with false</name>
              <secured>false</secured>
              <value>fake password</value>
            </WebAppAuthFormRecordField>
          </set>
        </fields>
      </formRecord>
      <comments>
        <set>
          <Comment><contents><![CDATA[some
comments]]></contents></Comment>
        </set>
      </comments>
  </WebAppAuthRecord>
  </data>
</ServiceRequest>
```

Example 3: Create - Selenium script (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/webappauthre
cord/" < file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
  <WebAppAuthRecord>
  <name><![CDATA[SELENIUM auth]]></name>
    <formRecord>
      <type>SELENIUM</type>
        <seleniumScript>
          <name><![CDATA[SS]]></name>
          <data><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">
<head profile="http://selenium-ide.openco.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
/>
<link rel="selenium.base" href="https://community.qualys.com/" />
<title>bnpparisbas-staging-bnptestad2</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">bnpparisbas-staging-
bnptestad2</td></tr>
</thead><tbody>
<tr>
<td>open</td>
<td>https://ssoforms-us-
uat.bnpparibas.com/cib/forms/login.fcc?TYPE=33554433&REALMOID=06-
7fe3816c-3fbf-115e-8bf1-
839478260cb3&GUID=&SMAUTHREASON=0&METHOD=GET&SMAGENTNAME=-SM-
R%2bgds%2bPTHEo7mSM%2fpGkcvnnTMJpZUZ1oJss49e%2fvYniGjw8bBdc5KSIU3W
oxRsmg&TARGET=-SM-HTTPS%3a%2f%2fstaging%2eprimebroker

%2ecom%2f</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>name=USER</td>
<td>bnptestad2</td>
```

```
</tr>
<tr>
<td>type</td>
<td>name=PASSWORD</td>
<td>S3cur1tyr0ck$</td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=ssoButton</td>
<td></td>
</tr>

</tbody></table>
</body>
</html>]]></data>
          <regex><![CDATA[^tri.*?]]></regex>
      </seleniumScript>
    </formRecord>
    <comments>
      <set>
        <Comment><contents><![CDATA[some
comments]]></contents></Comment>
      </set>
    </comments>
  </WebAppAuthRecord>
  </data>
</ServiceRequest>
```

Example 3: Create - server authentication (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/webappauthre
cord/" < file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
  <WebAppAuthRecord>
    <name><![CDATA[server auth]]></name>
      <serverRecord>
        <sslOnly>true</sslOnly>
        <certificate>
          <name><![CDATA[My Certificate]]></name>
<contents><![CDATA[-----BEGIN CERTIFICATE-----
MIIC4jCCAkugAwIBAgIJAPU+Kw6GX2aMMA0GCSqGSIb3DQEBBQUAMIGJMQswCQYD
VQQGEwJGUjEPMA0GA1UECAwGRnJhbmNlMREwDwYDVQQHDAhUb3Vsb3VzZTEPMA0G
A1UECgwGUXVhbHlzMRUwEwYDVQQLDAxRdWFseXMgVGVjaC4xDTALBgNVBAMMBE5p
Y28xHzAdBgkqhkiG9w0BCQEWEG5iaXplQHF1YWx5cy5jb20wHhcNMTExMDA1MjIx
...
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQC4SiB/HaNxQtwQUtot867MxTP1PqAQh7VyHIdBs037eafpd8B6
apHhih0Jw0zr2RzcWniUUhhpvwL4apG470/RzkIKSNu4h9akHqA5b0Pe0ZasrE7B
MxUZWNf9dfrY+JXQmdaPce0i4w4zZR+PabXDy5Mg9ONEUKS3AONCHk7acwIDAQAB
AoGAMHwAFLFdgLzQXNMPZ6uGv4TaaJkzT2YEzKLIyvY7e//Dt160GwDSpH3Lqffh
...
-----END RSA PRIVATE KEY-----]]></contents>
          <passphrase>My Certificate</passphrase>
        </certificate>
      </serverRecord>
    <comments>
      <set>
        <Comment><contents><![CDATA[some
comments]]></contents></Comment>
      </set>
    </comments>
  </WebAppAuthRecord>
  </data>
</ServiceRequest>
```

Update an authentication record

Update an authentication record which is in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/update/was/webappauthrecord/<id>

Methods allowed: POST

Input

The element "id" (Integer) is required, where "id" identifies an authentication record.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
The authentication record must be within the user's scope

Example

Update authentication record settings (POST)

Update the settings for authentication record ID 82605.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/webappauthre
cord/82605" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
    <WebAppAuthRecord>
      <name><![CDATA[Form and Server Auth]]></name>
      <serverRecord>
        <sslOnly>true</sslOnly>
        <fields>
          <set>
            <WebAppAuthServerRecordField>
              <type>DIGEST</type>
```

```
            <domain>realm</domain>
            <username><![CDATA[u]]></username>
            <password>p</password>
          </WebAppAuthServerRecordField>
        </set>
      </fields>
    </serverRecord>
    <formRecord>
      <type>STANDARD</type>
      <sslOnly>true</sslOnly>
      <fields>
        <set>
          <WebAppAuthFormRecordField>
            <name>username</name>
            <value>Login</value>
          </WebAppAuthFormRecordField>
        </set>
      </fields>
    </formRecord>
  </WebAppAuthRecord>
</data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebAppAuthRecord>
      <id>82605</id>
    </WebAppAuthRecord>
  </data>
</ServiceResponse>
```

Delete authentication records

Delete an authentication record which is in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/delete/was/webappauthrecord/<id>
or
https://qualysapi.qualys.com/qps/rest/3.0/delete/was/webappauthrecord

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

id (Integer)	lastScanDate (Date)
name (Text)	lastScan.authStatus (Text)
tags	isUsed (Boolean)
createdDate (Date)	contents
updateDate (Date)	

Permissions

User must have the WAS application enabled
User must have "API Access" permission
The authentication record must be within the user's scope

Examples

Example 1: Delete specific record (POST)

Delete authentication record ID 78149.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/webappauthre
cord/78149"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webappauthrecord.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebAppAuthRecord>
      <id>78149</id>
    </WebAppAuthRecord>
  </data>
</ServiceResponse>
```

Example 2: Delete multiple records (POST)

Delete authentication records that have a name containing the term "server".

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/webappauthre
cord/" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
    <filters>
        <Criteria field="name"
operator="CONTAINS">server</Criteria>
    </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
```

```
  <data>
    <WebAppAuthRecord>
      <id>12874</id>
    <WebAppAuthRecord>
    <WebAppAuthRecord>
      <id>13093</id>
    <WebAppAuthRecord>
  </data>
</ServiceResponse>
```

CHAPTER 4

Scan API

The WAS Scan API provides a suite of API functions for managing web application scans.

These operations are available:

Current scan count

Search scans

Get scan details

Launch a new scan

Retrieve the status of a scan

Retrieve the results of a scan

Cancel an unfinished scan

Delete an existing scan

Looking for something else?

Reference: WasScan

Reference: WAS Scan Results (legacy)

Current scan count

Returns the total number of scans in the user's account. Input elements are optional and are used to filter the number of scans included in the count.

URL: https://qualysapi.qualys.com/qps/rest/3.0/count/was/wasscan

Methods allowed: GET, POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: WasScan for descriptions of these <WasScan> elements.

id (Integer)	launchedDate (Date)
name (Text)	type (Keyword: DISCOVERY or VULNERABILITY)
webApp.name (Text)	mode (Keyword: MANUAL, SCHEDULED or API)
webApp.id (Integer)	status (Keyword: SUBMITTED, RUNNING, FINISHED, ERROR or CANCELED)
webApp.tags (with operator="NONE")	authStatus (Keyword: NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL)
webApp.tags.id (Integer)	resultsStatus (Keyword: NOT_USED, TO_BE_PROCESSED, NO_HOST_ALIVE, NO_WEB_SERVICE, TIME_LIMIT_EXCEEDED, SCAN_RESULTS_INVALID, SUCCESSFUL, PROCESSING
reference (Text)	

Allowed Operators	
Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN
Boolean	(true/false) EQUALS, NOT EQUALS

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Count includes scans on web applications within the user's scope

Examples

Example 1: Count - no criteria (GET)

Return a count of all scans in the user's account.

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/wasscan"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>534</count>
</ServiceResponse>
```

Example 2: Count - criteria (POST)

Return a count of scans that match all the criteria defined in the request POST data: 1) scan name contains the word "Schedule", 2) scan type is "VULNERABILITY", 3) the scanned web application contains the word "Merchant", and 4) the scan status is equal to "FINISHED".

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/wasscan" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
    <filters>
      <Criteria field="name"
operator="CONTAINS">Schedule</Criteria>
      <Criteria field="type"
operator="EQUALS">VULNERABILITY</Criteria
      <Criteria field="webApp.name"
operator="CONTAINS">Merchant</Criteria>
      <Criteria field="status"
operator="EQUALS">FINISHED</Criteria>
    </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
</ServiceResponse>
```

Example 3: Count - web applications without tags (POST)

Return a count of scans of web applications that do not have any tags assigned.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/wasscan" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <filters>
    <Criteria field="webApp.tags" operator="NONE"></Criteria>
  </filters>
</ServiceRequest>
```

Example 4: Count - web applications with certain tags (POST)

Return a count of scans of web applications that have certain tags assigned.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/wasscan" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <filters>
    <Criteria field="webApp.tags.id"
operator="EQUALS">1516928</Criteria>
 <Criteria field="webApp.tags.id"
operator="EQUALS">1234567</Criteria>
  </filters>
</ServiceRequest>
```

Search scans

Returns a list of scans on web applications which are in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/search/was/wasscan

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: WasScan for descriptions of these <WasScan> elements.

id (Integer)	launchedDate (Date)
name (Text)	type (Keyword: DISCOVERY or VULNERABILITY)
webApp.name (Text)	mode (Keyword: MANUAL, SCHEDULED or API)
webApp.id (Integer)	status (Keyword: SUBMITTED, RUNNING, FINISHED, ERROR or CANCELED)
webApp.tags (with operator="NONE")	authStatus (Keyword: NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL)
webApp.tags.id (Integer)	resultsStatus (Keyword: NOT_USED, TO_BE_PROCESSED, NO_HOST_ALIVE, NO_WEB_SERVICE, TIME_LIMIT_EXCEEDED, SCAN_RESULTS_INVALID, SUCCESSFUL, PROCESSING
reference (Text)	

Allowed Operators	
Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN
Boolean	(true/false) EQUALS, NOT EQUALS

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Output includes scans on web applications within the user's scope

Examples

Example 1: List running scans (POST)

Return a list of all running scans in the user's account.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/wasscan" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
   <filters>
       <Criteria field="status"
operator="EQUALS">RUNNING</Criteria>
    </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
```

```
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.cm/qps/xsd
/3.0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <WasScan>
      <id>13101</id>
      <name><![CDATA[Vulnerability Scan - 2011-02-24]]></name>
      <reference>was/1298538355659.20994</reference>
      <type>VULNERABILITY</type>
      <mode>ONDEMAND</mode>
      <profile>
        <id>1072</id>
        <name><![CDATA[Initial WAS Options]]></name>
      </profile>
      <launchedDate>2011-02-24T10:05:55Z</launchedDate>
      <launchedBy>
        <id>123056</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </launchedBy>
      <status>RUNNING</status>
    </WasScan>
    <WasScan>
      <id>13102</id>
      <name><![CDATA[Vulnerability Scan - 2011-02-24]]></name>
      <reference>was/1298541157873.20995</reference>
      <type>VULNERABILITY</type>
      <mode>ONDEMAND</mode>
      <profile>
        <id>1072</id>
        <name><![CDATA[Initial WAS Options]]></name>
      </profile>
      <launchedDate>2011-02-24T10:52:37Z</launchedDate>
      <launchedBy>
        <id>123056</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
```

```
      </launchedBy>
      <status>RUNNING</status>
    </WasScan>
  </data>
</ServiceResponse>
```

Example 2: List scans - successful authentication (POST)

Return a list of scans in the user's account that successfully authenticated to the target web application.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X “POST”
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/wasscan" <
file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
   <filters>
       <Criteria field="authStatus"
operator="EQUALS">SUCCESSFUL</Criteria>
   </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <WasScan>
      <id>13096</id>
      <name><![CDATA[Web Vulnerability Scan - 2011-02-23]]></name>
      <reference>was/1298475533625.20931</reference>
      <type>VULNERABILITY</type>
      <mode>ONDEMAND</mode>
```

```
      <profile>
        <id>1072</id>
        <name><![CDATA[Initial WAS Options]]></name>
      </profile>
      <launchedDate>2011-02-23T16:38:53Z</launchedDate>
      <launchedBy>
        <id>123056</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </launchedBy>
      <status>FINISHED</status>
    </WasScan>
    <WasScan>
      <id>13116</id>
      <name><![CDATA[Relaunch Vulnerability Scan - 2011-02-
23]]></name>
      <reference>was/1298558684177.21009</reference>
      <type>VULNERABILITY</type>
      <mode>ONDEMAND</mode>
      <profile>
        <id>1072</id>
        <name><![CDATA[Initial WAS Options]]></name>
      </profile>
      <launchedDate>2011-02-24T15:44:44Z</launchedDate>
      <launchedBy>
        <id>123056</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </launchedBy>
      <status>FINISHED</status>
    </WasScan>
  </data>
</ServiceResponse>
```

Example 3: List scans - web applications without tags (POST)

Return a list of scans of web applications that do not have any tags assigned.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/wasscan" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <filters>
    <Criteria field="webApp.tags" operator="NONE"></Criteria>
  </filters>
</ServiceRequest>
```

Example 3: List scans - web applications with certain tags (POST)

Return a list of scans of web applications that have certain tags assigned.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/wasscan" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <filters>
    <Criteria field="webApp.tags.id"
operator="EQUALS">1516928</Criteria>
 <Criteria field="webApp.tags.id"
operator="EQUALS">1234567</Criteria>
  </filters>
</ServiceRequest>
```

Get scan details

View details for a scan on a web application which is in the user's scope. Want to find a scan ID to use as input? See "Search scans".

URL: https://qualysapi.qualys.com/qps/rest/3.0/get/was/wasscan/<id>

Methods allowed: GET

Input

The element "id" (Integer) is required, where "id" identifies a scan.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Scan target must be within the user's scope

Example

Example 1: List scan details (GET)

View details for the scan with the ID 1275177.

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/wasscan/1275177
"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/scan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>1275177</id>
      <name><![CDATA[My Vulnerability Scan]]></name>
      <reference>was/1435351699783.1836970</reference>
```

```
<type>VULNERABILITY</type>
<mode>API</mode>
<progressiveScanning>true</progressiveScanning>
<multi>false</multi>
<target>
  <webApp>
    <id>2376280</id>
    <name><![CDATA[My Web App]]></name>
    <url><![CDATA[http://10.10.26.238]]></url>
  </webApp>
  <scannerAppliance>
    <type>EXTERNAL</type>
  </scannerAppliance>
  <cancelOption>DEFAULT</cancelOption>
</target>
<profile>
  <id>332147</id>
  <name><![CDATA[10 links]]></name>
</profile>
<options>
  <count>14</count>
  <list>
    <WasScanOption>
     <name>Web Application Authentication Record Name</name>
      <value><![CDATA[None]]></value>
    </WasScanOption>
    <WasScanOption>
      <name>Detection Scope</name>
      <value><![CDATA[COMPLETE]]></value>
    </WasScanOption>
    <WasScanOption>
      <name>Scanner Appliance</name>
      <value><![CDATA[External]]></value>
    </WasScanOption>
    <WasScanOption>
      <name>Target URL</name>
      <value><![CDATA[http://10.10.26.238]]></value>
    </WasScanOption>
    <WasScanOption>
      <name>Performance Settings</name>
      <value><![CDATA[LOW]]></value>
```

```
            </WasScanOption>
            <WasScanOption>
              <name>Sensitive Content: Social Security Numbers
(US)</name>
              <value><![CDATA[false]]></value>
            </WasScanOption>
            <WasScanOption>
              <name>Sensitive Content: Credit Card Numbers</name>
              <value><![CDATA[false]]></value>
            </WasScanOption>
            <WasScanOption>
              <name>Maximum Crawling Links</name>
              <value><![CDATA[10]]></value>
            </WasScanOption>
            <WasScanOption>
              <name>Bruteforce Settings</name>
              <value><![CDATA[MINIMAL]]></value>
            </WasScanOption>
            <WasScanOption>
              <name>Option Profile Name</name>
              <value><![CDATA[10 links]]></value>
            </WasScanOption>
            <WasScanOption>
              <name>Crawling Form Submissions</name>
              <value><![CDATA[BOTH]]></value>
            </WasScanOption>
            <WasScanOption>
              <name>Request Parameter Set</name>
              <value><![CDATA[Initial Parameters]]></value>
            </WasScanOption>
            <WasScanOption>
              <name>Cancel After N Hours</name>
              <value><![CDATA[2]]></value>
            </WasScanOption>
            <WasScanOption>
              <name>Web Application Name</name>
              <value><![CDATA[My Web App]]></value>
            </WasScanOption>
          </list>
        </options>
        <launchedDate>2015-06-26T20:48:19Z</launchedDate>
```

```
      <launchedBy>
        <id>2086786</id>
        <username>acme_tp16</username>
        <firstName><![CDATA[FIRSTNAME]]></firstName>
        <lastName><![CDATA[LASTNAME]]></lastName>
      </launchedBy>
      <status>RUNNING</status>
    </WasScan>
  </data>
</ServiceResponse>
```

Launch a new scan

Launch a scan on a web application which is in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/launch/was/wasscan

Methods allowed: POST

Input

Required input elements are listed below. The associated data type for each element appears in parentheses, unless a compound element. See Reference: WasScan for descriptions of these <WasScan> elements.

Required Elements	Optional Elements
name (Text)	scannerAppliance
webApp.id (Integer)	webAppAuthRecord.id (Integer)
type (Keyword: DISCOVERY or VULNERABILITY)	options
profile.id (Integer)*	proxy.id (Integer)
	cancelOption set to DEFAULT - Forces the use of the target web app's cancelScans option if set, else fall back to the one passed in to the API while launching the scan
	cancelOption set to SPECIFIC - Always use the cancel scan option passed while launching the scan

* The element profile (Text) is required unless the target has a default option profile.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Launch WAS Scan" permission
Scan target must be within the user's scope

Examples

Example 1: Launch a new scan - basic elements

Launch a new discovery scan on the web application ID 323126 using the option profile ID 1021, and cancel the scan after 5 hours.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/launch/was/wasscan" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
    <WasScan>
      <name>New WAS Discovery Scan launched from API</name>
      <type>DISCOVERY</type>
      <target>
        <webApp>
          <id>323126</id>
        </webApp>
        <scannerAppliance>
          <type>EXTERNAL</type>
        </scannerAppliance>
      </target>
      <profile>
        <id>1021</id>
      </profile>
      <options>
        <WasScanOption>
          <name>Cancel After N Hours</name>
          <value><![CDATA[5]]></value>
          </WasScanOption>
      </options>
    </WasScan>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>16954</id>
    </WasScan>
  </data>
</ServiceResponse>
```

Example 2: Launch a new scan - use proxy

Launch a new vulnerability scan using proxy ID 12345.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/launch/was/wasscan" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
    <WasScan>
      <name>New WAS Vulnerability Scan launched from API</name>
      <type>VULNERABILITY</type>
      <target>
        <webApp>
          <id>323126</id>
        </webApp>
        <scannerAppliance>
          <type>Internal</type>
          <friendlyName>dp_scanner</friendlyName>
        </scannerAppliance>
        <proxy>
          <id>12345</id>
        </proxy>
```

```
      </target>
      <profile>
        <id>1021</id>
      </profile>
    </WasScan>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>224466</id>
    </WasScan>
  </data>
</ServiceResponse>
```

Example 3: Launch a new scan - set cancel scan option

Launch a new vulnerability scan on web app ID 2376280 and set the cancel scan option to DEFAULT. This forces the use of the target web app's cancelScans option if set.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/launch/was/wasscan" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
    <WasScan>
      <name><![CDATA[My Vulnerability Scan]]></name>
      <type>VULNERABILITY</type>
      <target>
        <webApp>
          <id>2376280</id>
        </webApp>
        <scannerAppliance>
          <type>EXTERNAL</type>
        </scannerAppliance>
        <cancelOption>DEFAULT</cancelOption>
      </target>
    </WasScan>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>1275177</id>
    </WasScan>
  </data>
</ServiceResponse>
```

Retrieve the status of a scan

Retrieve the status of a scan on a web application which is in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/status/was/wasscan/<id>

Methods allowed: GET

Input

The element "id" (Integer) is required, where "id" identifies a scan.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Scan target must be within the user's scope

Example

Get the status of the scan with the ID 164.

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/status/was/wasscan/164"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>164</id>
      <status>SUBMITTED</status>
    </WasScan>
  </data>
</ServiceResponse>
```

Retrieve the results of a scan

Retrieve the results of a scan on a web application which is in the user's scope. Include "3.0" in the URL for WASA v3 scan results using the WAS API schema, part of the API V3 architecture (see https://qualysapi.qualys.com/qps/xsd/3.0/was/wasscan.xsd). Include "2.0" in the URL for scan results in legacy format (WAS v2 and earlier), using the webapp_scan.dtd - see Reference: WAS Scan Results (legacy).

URL: https://qualysapi.qualys.com/qps/rest/3.0/download/was/wasscan/<id>
or
https://qualysapi.qualys.com/qps/rest/2.0/download/was/wasscan/<id>

Methods allowed: GET

Tip When you download web application scan results using the WAS API, you'll want to view vulnerability descriptions from the Qualys KnowledgeBase in order to understand the vulnerabilities detected and see our recommended solutions. See How to Download Vulnerability Details (in Chapter 1).

Input

The element "id" (Integer) is required, where "id"identifies a scan.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Scan target must be within the user's scope

Example

Download results of a scan

Download the results of the scan with the ID 174726.

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/download/was/wasscan/174726"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<WasScan xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
  <id>174726</id>
  <name><![CDATA[My Web Application Scan]]></name>
  <reference>was/1328563860860.218807</reference>
  <type>VULNERABILITY</type>
  <mode>API</mode>
  <target>
    <webApp>
      <id>952835</id>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[https://example.com/]]></url>
    </webApp>
    <scannerAppliance>
      <type>INTERNAL</type>
      <friendlyName><![CDATA[is_quays_tc321]]></friendlyName>
    </scannerAppliance>
  </target>
  <profile>
    <id>6714</id>
    <name><![CDATA[Initial WAS Options]]></name>
  </profile>
  <options>
    <count>10</count>
    <list>
      <WasScanOption>
        <name>Detection Scope</name>
        <value>COMPLETE</value>
      </WasScanOption>
      <WasScanOption>
        <name>Maximum Crawling Links</name>
        <value>300</value>
      </WasScanOption>
      <WasScanOption>
        <name>Bruteforce Settings</name>
        <value>MINIMAL</value>
      </WasScanOption>
      <WasScanOption>
```

```
          <name>Option Profile Name</name>
          <value>Initial WAS Options</value>
        </WasScanOption>
       <WasScanOption>
          <name>Scanner Appliance Name</name>
         <value><![CDATA[External (IP: 10.40.3.104, Scanner: 6.2.13-
         1, WAS: 2.13.5-1, Signatures: 2.2.52-2)]]></value>
        </WasScanOption>
       <WasScanOption>
          <name>Ignore Binary Files</name>
          <VALUE><![CDATA[true]]></VALUE>
       </WasScanOption>

...

    </list>
  </options>
  <launchedDate>2012-02-06T21:31:00Z</launchedDate>
  <launchedBy>
    <id>35842</id>
    <username>username</username>
    <firstName><![CDATA[John]]></firstName>
    <lastName><![CDATA[Smith]]></lastName>
  </launchedBy>
  <status>FINISHED</status>
  <endScanDate>2012-02-06T21:49:34Z</endScanDate>
  <scanDuration>1114</scanDuration>
  <summary>
    <crawlDuration>16</crawlDuration>
    <testDuration>138</testDuration>
    <linksCollected>10</linksCollected>
    <linksCrawled>1</linksCrawled>
    <nbRequests>503</nbRequests>
    <averageResponseTime>0.001554</averageResponseTime>
    <resultsStatus>SUCCESSFUL</resultsStatus>
    <authStatus>NONE</authStatus>
  </summary>
  <stats>
    <global>
      <nbVulnsTotal>79</nbVulnsTotal>
      <nbVulnsLevel5>24</nbVulnsLevel5>
```

```
      <nbVulnsLevel4>0</nbVulnsLevel4>
      <nbVulnsLevel3>3</nbVulnsLevel3>
      <nbVulnsLevel2>18</nbVulnsLevel2>
      <nbVulnsLevel1>34</nbVulnsLevel1>
      <nbScsTotal>0</nbScsTotal>
      <nbScsLevel5>0</nbScsLevel5>
      <nbScsLevel4>0</nbScsLevel4>
      <nbScsLevel3>0</nbScsLevel3>
      <nbScsLevel2>0</nbScsLevel2>
      <nbScsLevel1>0</nbScsLevel1>
      <nbIgsTotal>10</nbIgsTotal>
      <nbIgsLevel5>0</nbIgsLevel5>
      <nbIgsLevel4>0</nbIgsLevel4>
      <nbIgsLevel3>0</nbIgsLevel3>
      <nbIgsLevel2>0</nbIgsLevel2>
      <nbIgsLevel1>10</nbIgsLevel1>
    </global>
    <byGroup>
      <count>3</count>
      <list>
        <GroupStat>
          <group>PATH</group>
          <nbTotal>18</nbTotal>
          <nbLevel5>0</nbLevel5>
          <nbLevel4>0</nbLevel4>
          <nbLevel3>0</nbLevel3>
          <nbLevel2>18</nbLevel2>
          <nbLevel1>0</nbLevel1>
        </GroupStat>

...

      </list>
    </byGroup>
    <byOwasp>
      <count>4</count>
      <list>
        <OwaspStat>
          <owasp>OWASP-A4</owasp>
          <nbTotal>18</nbTotal>
          <nbLevel5>0</nbLevel5>
```

```
            <nbLevel4>0</nbLevel4>
            <nbLevel3>0</nbLevel3>
            <nbLevel2>18</nbLevel2>
            <nbLevel1>0</nbLevel1>
          </OwaspStat>

...

      </list>
    </byOwasp>
    <byWasc>
      <count>5</count>
      <list>
        <WascStat>
          <wasc>WASC-15</wasc>
          <nbTotal>14</nbTotal>
          <nbLevel5>0</nbLevel5>
          <nbLevel4>0</nbLevel4>
          <nbLevel3>2</nbLevel3>
          <nbLevel2>12</nbLevel2>
          <nbLevel1>0</nbLevel1>
        </WascStat>

...

      </list>
    </byWasc>
  </stats>
  <vulns>
    <count>79</count>
    <list>
      <WasScanVuln>
        <qid>150081</qid>
        <title><![CDATA[Possible Clickjacking
vulnerability]]></title>

<uri><![CDATA[https://example.com/randomLink/1328558353.9231]]></u
ri>
        <instances>
          <count>1</count>
          <list>
            <WasScanVulnInstance>
```

```
              <authenticated>false</authenticated>
              <payloads>
                <count>1</count>
                <list>
                  <WasScanVulnPayload>
                    <payload><![CDATA["'>
<qss%20a=@REQUESTID@>]]></payload>
                    <result base64="true">
<![CDATA[c3RhcnQoKTogVGhlIHNlc3Npb24gaWQgY29udGFpbnMgaW52YWxpZCBja
GFyYWN0ZXJzLCB2YWxpZCBjaGFyYWN0ZXJzIGFyZSBvbmx5IGEteiwgQS1aIGFuZCA
wLTkgaW4gJmx0O2ImZ3Q7L3Zhci93d3cvaHRtbC9pbmNsdWRlcy9jb25maWcucGhwJ
mx0Oy9iJmd0OyBvbiBsaW5lICZsdDtiJmd0OzImbHQ7L2ImZ3Q7Jmx0O2JyIC8mZ3Q
7CiZsdDticiAvJmd0OwombHQ7YiZndDtXYXJuaW5nJmx0Oy9iJmd0OzogIHNlc3Npb
25fc3RhcnQoKTogQ2Fubm90IHNlbmQgc2Vzc2lvbiBjYWNoZSBsaW1pdGVyIC0gaGV
hZGVycyBhbHJlYWR5IHNlbnQgKG91dHB1dCBzdGFydGVkIGF0IC92YXIvd3d3L2h0b
WwvaW5jbHVkZXMvY29uZmlnLnBocDoyKSBpbiAmbHQ7YiZndDsvdmFyL3d3dy9odG1
sL2luY2x1ZGVzL2NvbmZpZy5waHAmbHQ7L2ImZ3Q7IG9uIGxpbmUgJmx0O2ImZ3Q7M
iZsdDsvYiZndDsmbHQ7YnIgLyZndDsKJmx0O2JyIC8mZ3Q7CiZsdDtiJmd0O1dhcm5
pbmcmbHQ7L2ImZ3Q7OiAgQ2Fubm90IG1vZGlmeSBoZWFkZXIgaW5mb3JtYXRpb24gL
SBoZWFkZXJzIGFscmVhZHkgc2VudCBieSAob3V0cHV0IHN0YXJ0ZWQgYXQgL3Zhci9
3d3cvaHRtbC8]]></result>
                  </WasScanVulnPayload>
                </list>
              </payloads>
            </WasScanVulnInstance>
          </list>
        </instances>
      </WasScanVuln>

...


    </list>
  </vulns>
  <sensitiveContents>
    <count>0</count>
  </sensitiveContents>
  <igs>
    <count>10</count>
    <list>
      <WasScanIg>
        <qid>150058</qid>
```

```
        <title><![CDATA[Flash Analysis]]></title>
        <data base64="true"><![CDATA
[U1dGIGZpbGU6IGh0dHA6Ly8xMC4xMC4yNi4yMzg6ODAvYm9xL2FjY3QvcGVyc29uY
Wwvd2ludGVyMi5zd2YKICAgICBWZXJzaW9uOiA4CgpTV0YgZmlsZTogaHR0cDovLzE
wLjEwLjI2LjIzOC9ib3EvcHJvdGVjdGVkL21pbWUvZGVmYXVsdFBhZ2Uuc3dmCiAgI
CAgVmVyc2lvbjogNgoK]]></data>
      </WasScanIg>

...

    </list>
  </igs>
</WasScan>
```

Cancel an unfinished scan

Cancel an unfinished scan on a web application which is in the user's scope. Note that scan results will not be returned.

URL: https://qualysapi.qualys.com/qps/rest/3.0/cancel/was/wasscan/<id>

Methods allowed: POST

Input

The element "id" (Integer) is required, where "id" identifies a running scan.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Cancel WAS Scan" permission

Example

Cancel the unfinished scan that has the ID 168.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/cancel/was/wasscan/168"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>168</id>
    </WasScan>
  </data>
</ServiceResponse>
```

Delete an existing scan

Delete an existing scan on a web application which is in the user's scope. You can delete any scan in your account that is not running.

URL: https://qualysapi.qualys.com/qps/rest/3.0/delete/was/wasscan/<id>
or
https://qualysapi.qualys.com/qps/rest/3.0/delete/was/wasscan

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: WasScan for descriptions of these <WasScan> elements.

id (Integer)	type (Keyword: DISCOVERY or VULNERABILITY)
name (Text)	mode (Keyword: MANUAL, SCHEDULED or API)
webApp.name (Text)	status (Keyword: SUBMITTED, RUNNING, FINISHED, ERROR or CANCELED)
webApp.id (Integer)	authStatus (Keyword: NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL)
reference (Text)	resultsStatus (Keyword: NOT_USED, TO_BE_PROCESSED, NO_HOST_ALIVE, NO_WEB_SERVICE, TIME_LIMIT_EXCEEDED, SCAN_RESULTS_INVALID, SUCCESSFUL, PROCESSING
launchedDate (Date)	

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Delete WAS scan" permission

Examples

Example 1: Delete a specified scan (POST)

Delete the scan with the ID 12405.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/wasscan/1240
5"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>12405</id>
    </WasScan>
  </data>
</ServiceResponse>
```

Example 2: Delete scans with criteria (POST)

Delete scans with a name that contains the string "VULN".

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/wasscan" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
    <filters>
        <Criteria field="name" operator="CONTAINS">VULN</Criteria>
```

```
    </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
  <data>
    <WasScan>
      <id>12874</id>
    </WasScan>
    <WasScan>
      <id>13093</id>
    </WasScan>
  </data>
</ServiceResponse>
```

Reference: WasScan

The <WasScan> element includes sub elements used to define a web application scan. A reference of these elements is provided below. An asterisk * indicates a complex element.

Element (data type)	Description
id (Integer)	The scan ID. This element is assigned by the service and is required for a certain type of request (details, status, results or cancel).
name (Text)	The user-defined scan name (maximum 256 characters).

target*	The target of the scan. The target includes the web application and authentication records, if any.
	
	`<scannerAppliance>` - type (Keyword) is set to INTERNAL for a scanner appliance, or EXTERNAL for external scanners. If the type is INTERNAL, friendlyName (Text) is the user-defined appliance name.
	
	`<cancelOption>` set to DEFAULT - Forces the use of the target web app's cancelScans option if set, else fall back to the one passed in to the API while launching the scan.
	
	`<cancelOption>` set to SPECIFIC - Always use the cancel scan option passed while launching the scan.
	
	Example:
	`<target>`
	`  <webApp>`
	`  <id>323126</id>`
	`  </webApp>`
	`  <webAppAuthRecord>`
	`  <id>1054</id>`
	`  </webAppAuthRecord>`
	`  <scannerAppliance>`
	`  <type>Internal</type>`
	`  <friendlyName>dp_scanner</friendlyName>`
	`  </scannerAppliance>`
	`  <cancelOption>DEFAULT</cancelOption>`
	`</target>`

type (Keyword)	The scan type: VULNERABILITY or DISCOVERY.

Element (data type)	Description

profile.id (Integer))	The name of the option profile that includes scan settings. The service provides the profile "Initial WAS Options" and we recommend this to get started.
	Example:
	`<profile>`
	`  <name>Initial WAS Options</name>`
	`</profile>`

proxy.id (Integer)	The proxy for scanning the target web application.
	Example:
	`<proxy>`
	`  <id>12345</id>`
	`</proxy>`

options	The cancelAfterNHours element defines a number of hours after which a running scan will be canceled.
	`<options>`
	`  <WasScanOption>`
	`  <name>Cancel After N Hours</name>`
	`  <value><![CDATA[1]]></value>`
	`  </WasScanOption>`
	`</options>`
	The cancelTime element defines a specific time at which a running scan will be canceled.
	Example:
	`<cancelTime>11:15</cancelTime>`

Scanner Appliance	The IP address of the external scanner appliance, when an external scanner is used.
mode (Keyword)	The mode of the scan: ONDEMAND, SCHEDULED or API.
launchedDate (Date)	The date and time when the scan was launched in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).

launchedBy*	The user who launched the scan. User properties include user ID, user login, first and last name.
	Example:
	`<launchedBy>`
	`  <id>123056</id>`
	`  <username>username</username>`
	`  <firstName><![CDATA[John]]></firstName>`
	`  <lastName><![CDATA[Smith]]></lastName>`
	`</launchedBy>`

status (Keyword)	The status of the scan: SUBMITTED, RUNNING, FINISHED, ERROR or CANCELED.
endScanDate (Date)	The date and time when the scan ended in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).

Element (data type)	Description

summary*	The scan summary. <crawlTime> is the length of time used to crawl the web application. <testDuration> is the length of time used to perform analysis. <nbRequests> is the number of requests sent during the scan. <authStatus> is the authentication status (NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL)
	Example:
	`<summary>`
	`  <crawlTime>22.0</crawlTime>`
	`  <testDuration>112.0</testTime>`
	`  <linksCrawled>17</linksCrawled>`
	`  <nbRequests>3814</nbRequests>`
	`  <os>Windows XP SP2</os>`
	`<resultsStatus>RESULTS_PROCESSED_SUCCESSFULLY</resultsStatus>`
	`  <authStatus>NO_AUTH</authStatus>`
	`</summary>`

vulns*	The list of detected vulnerabilities. Each <WasScanVuln> element identifies a particular vulnerability QID and the URI where detected, each <WasScanVulnInstance> element identifies a vulnerability instance, and each <WasScanVulnInstancePayload> element identifies associated payloads.
igs*	The detected information gathered. Each <WasScanIg> element identifies a particular information gathered QID.
sensitiveContents*	The detected sensitive content. Each <WasScanSensitiveContent> element identifies a particular sensitive content QID and the URI where detected, each <instances> element identifies a sensitive content instance, and each <WasScanSensitiveContentInstancePayLoad> element identifies associated payloads.
stats*	The statistics gathered by the scan: the total number of vulnerabilities, the number of vulnerabilities by severity level, information gathered by severity level and the number of vulnerabilities by group, OWASP and WASC.

Reference: WAS Scan Results (legacy)

You have the option to retrieve web application scan results in legacy format (WAS v2 and earlier), using the webapp_scan.dtd (see "Retrieve the results of a scan"). You can download this DTD by going to https://qualysapi.qualys.com/webapp_scan.dtd (where qualysapi is the API server URL where your account is located).

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- QUALYS WEB APPLICATION SCAN DTD -->

<!ELEMENT WEB_APPLICATION_SCAN (ERROR | (HEADER, SUMMARY,
                                RESULTS))>
<!ELEMENT ERROR (#PCDATA)>
<!ATTLIST ERROR number CDATA #IMPLIED>

<!-- GENERIC HEADER -->
<!ELEMENT HEADER (NAME, GENERATION_DATETIME, COMPANY_INFO,
                  USER_INFO)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT GENERATION_DATETIME (#PCDATA)>

<!ELEMENT COMPANY_INFO (NAME, ADDRESS, CITY, STATE, COUNTRY,
                        ZIP_CODE)>
<!ELEMENT ADDRESS (#PCDATA)>
<!ELEMENT CITY (#PCDATA)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT COUNTRY (#PCDATA)>
<!ELEMENT ZIP_CODE (#PCDATA)>

<!ELEMENT USER_INFO (NAME, USERNAME, ROLE)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT ROLE (#PCDATA)>

<!-- SUMMARY -->
<!ELEMENT SUMMARY (SCAN_SUMMARY, VULN_SUMMARY?,
                   SENSITIVE_CONTENT_SUMMARY)>
<!ELEMENT SCAN_SUMMARY (SCAN_INFO*)>
<!ELEMENT SCAN_INFO (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>

<!ELEMENT VULN_SUMMARY (VULN_GROUP*)>
```

```
<!ELEMENT VULN_GROUP (TITLE, SEVERITY_5, SEVERITY_4, SEVERITY_3,
                      SEVERITY_2, SEVERITY_1, TOTAL)>
<!ELEMENT SEVERITY_1 (#PCDATA)>
<!ELEMENT SEVERITY_2 (#PCDATA)>
<!ELEMENT SEVERITY_3 (#PCDATA)>
<!ELEMENT SEVERITY_4 (#PCDATA)>
<!ELEMENT SEVERITY_5 (#PCDATA)>
<!ELEMENT TOTAL (#PCDATA)>

<!ELEMENT SENSITIVE_CONTENT_SUMMARY (SENSITIVE_CONTENT_GROUP*)>
<!ELEMENT SENSITIVE_CONTENT_GROUP (TITLE, TOTAL)>

<!-- RESULTS -->
<!ELEMENT RESULTS (VULN_LIST?, SENSITIVE_CONTENT_LIST?,
                   INFO_LIST?)>

<!ELEMENT VULN_LIST (VULN*)>
<!ELEMENT VULN (GROUP, QID, TITLE, VULN_INSTANCES)>
<!ELEMENT VULN_INSTANCES (VULN_INSTANCE*)>
<!ELEMENT VULN_INSTANCE (HOST, PORT, URI, AUTHENTICATED?,
FORM_ENTRY_POINT?, PARAMS, FINDINGS)>
<!ELEMENT AUTHENTICATED (#PCDATA)>
<!ELEMENT FORM_ENTRY_POINT (#PCDATA)>

<!ELEMENT SENSITIVE_CONTENT_LIST (SENSITIVE_CONTENT*)>
<!ELEMENT SENSITIVE_CONTENT (GROUP, QID, TITLE,
                             SENSITIVE_CONTENT_INSTANCES)>
<!ELEMENT SENSITIVE_CONTENT_INSTANCES
(SENSITIVE_CONTENT_INSTANCE*)>
<!ELEMENT SENSITIVE_CONTENT_INSTANCE (HOST, PORT, URI, CONTENT?,
                                      FINDINGS)>

<!ELEMENT INFO_LIST (INFO*)>
<!ELEMENT INFO (QID, TITLE, RESULT)>

<!ELEMENT GROUP (#PCDATA)>
<!ELEMENT QID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT HOST (#PCDATA)>
<!ELEMENT PORT (#PCDATA)>
```

```
<!ELEMENT URI (#PCDATA)>
<!ELEMENT CONTENT (#PCDATA)>
<!ELEMENT PARAMS (#PCDATA)>
<!ELEMENT FINDINGS (FINDING*)>
<!ELEMENT FINDING (PAYLOAD?, RESULT)>
<!ELEMENT PAYLOAD (#PCDATA)>
<!ELEMENT RESULT (#PCDATA)>
<!ATTLIST RESULT base64 (true|false) "false">
```

CHAPTER 5

Schedule API

The WAS Schedule API provides a suite of API functions for managing web application scan schedules across the enterprise.

These operations are available:

Current schedule count

Search schedules

Get schedule details

Create a schedule

Update a schedule

Activate an existing schedule

Deactivate an existing schedule

Delete one or more existing schedules

Download one or more schedules to iCalendar

Current schedule count

Returns the total number of schedules in the user's account. Input elements are optional and are used to filter the number of schedules included in the count.

URL: https://qualysapi.qualys.com/qps/rest/3.0/count/was/wasscanschedule

Methods allowed: GET, POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: WasScanSchedule for descriptions of these <WasScanSchedule> elements

id (Integer)	webApp.name (Text)
name (Text)	webApp.id (Integer)
owner.id (Text)	webApp.tags (with operator="NONE")
createdDate (Date)	webApp.tags.id (Integer)
updatedDate (Date)	active (Boolean)
type (Keyword: DISCOVERY or VULNERABILITY)	invalid (Boolean)

Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN
Boolean	(true/false) EQUALS, NOT EQUALS

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Scan target must be within the user's scope

Examples

Example 1: Count - no criteria (GET)

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/wasscanschedu
le"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>15</count>
</ServiceResponse>
```

Example 2: Count - criteria (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/wasscanschedu
le" < file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <filters>
    <Criteria field="type" operator="EQUALS">DISCOVERY</Criteria>
  </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscanschedule.xsd">
```

```
  <responseCode>SUCCESS</responseCode>
  <count>3</count>
</ServiceResponse>
```

Example 3: Count - web applications without tags (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/wasscanschedu
le" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <filters>
    <Criteria field="webApp.tags" operator="NONE"></Criteria>
  </filters>
</ServiceRequest>
```

Example 2: Count - web applications with certain tags (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/wasscanschedu
le" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <filters>
    <Criteria field="webApp.tags.id"
operator="EQUALS">1516928</Criteria>
 <Criteria field="webApp.tags.id"
operator="EQUALS">1234567</Criteria>
  </filters>
</ServiceRequest>
```

Search schedules

Returns a list of scheduled scans on web applications which are in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/search/was/wasscanschedule

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: WasScanSchedule for descriptions of these <WasScanSchedule> elements.

id (Integer)	webApp.tags (with operator="NONE")
name (Text)	webApp.tags.id (Integer)
owner.id	updatedDate (Date)
createdDate (Date)	invalid (Boolean)
active (Boolean)	lastScan (with operation="NONE")
type (Keyword: DISCOVERY or VULNERABILITY)	lastScan.launchedDate (Date)
webApp.name (Text)	lastScan.status (keyword: SUBMITTED, RUNNING, FINISHED, ERROR, CANCELLED)
webApp.id (Integer)	

Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN
Boolean	(true/false) EQUALS, NOT EQUALS

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Scan target must be within the user's scope

Examples

Example 1: List schedules never launched - criteria (GET)

Request:

```
curl -u "USERNAME:PASSWD"
https://qualysapi.qualys.com/qps/rest/3.0/search/was/wasscanschedu
le < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
        <filters>
                <Criteria field="lastScan"
operator="NONE"></Criteria>
        </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <WasScanSchedule>
      <id>171425669</id>
     <name><![CDATA[Web Application Vulnerability Scan - 2014-Aug-
19]]></name>
      <owner>
        <id>8792415669</id>
      </owner>
      <active>false</active>
```

```
      <type>VULNERABILITY</type>
      <target>
        <webApp>
          <id>1296335669</id>
          <name><![CDATA[My Web Application]]></name>
          <url><![CDATA[http://10.10.1.100]]></url>
        </webApp>
        <webAppAuthRecord>
          <id>175535669</id>
          <name><![CDATA[AR1]]></name>
        </webAppAuthRecord>
        <scannerAppliance>
          <type>EXTERNAL</type>
        </scannerAppliance>
      </target>
      <profile>
        <id>716315669</id>
        <name><![CDATA[Copy of Initial WAS Options]]></name>
      </profile>
      <scheduling>
        <startDate>2014-08-19T12:30:00Z</startDate>
        <timeZone>
          <code>America/Dawson</code>
          <offset>-07:00</offset>
        </timeZone>
        <occurrenceType>ONCE</occurrenceType>
      </scheduling>
      <createdDate>2014-08-19T19:30:49Z</createdDate>
      <updatedDate>2014-08-19T19:30:50Z</updatedDate>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Example 2: List launched schedules - criteria (GET)

Request:

```
curl -u "USERNAME:PASSWD"
https://qualysapi.qualys.com/qps/rest/3.0/search/was/wasscanschedu
le < file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
        <filters>
        <Criteria field="lastScan.status"
operator="IN">FINISHED,ERROR</Criteria>
        <Criteria field="lastScan.launchedDate"
operator="LESSER">2014-08-19</Criteria>
        </filters>
</ServiceRequest>
```

Response:

```
…
    </WasScanSchedule>
    <WasScanSchedule>
      <id>97354000</id>
      <name><![CDATA[Schedule Notification]]></name>
      <owner>
        <id>334527</id>
      </owner>
      <active>false</active>
      <type>VULNERABILITY</type>
      <target>
        <webApp>
          <id>1061764000</id>
          <name><![CDATA[My Web App]]></name>
          <url><![CDATA[http://10.10.26.238]]></url>
        </webApp>
        <webAppAuthRecord>
          <id>8753</id>
          <name><![CDATA[Auth Record 1]]></name>
        </webAppAuthRecord>
        <scannerAppliance>
          <type>EXTERNAL</type>
        </scannerAppliance>
      </target>
      <profile>
        <id>55784</id>
        <name><![CDATA[Initial WAS Options]]></name>
      </profile>
      <scheduling>
        <startDate>2014-05-06T18:22:00Z</startDate>
```

```
        <timeZone>
          <code>America/Dawson</code>
          <offset>-07:00</offset>
        </timeZone>
        <occurrenceType>DAILY</occurrenceType>
        <occurrence>
          <dailyOccurrence>
            <everyNDays>1</everyNDays>
          </dailyOccurrence>
        </occurrence>
      </scheduling>
      <lastScan>
        <id>14929668885</id>
        <launchedDate>2014-05-12T01:22:02Z</launchedDate>
        <status>FINISHED</status>
      </lastScan>
      <createdDate>2014-05-06T23:17:23Z</createdDate>
      <updatedDate>2014-05-13T01:22:02Z</updatedDate>
    </WasScanSchedule>
...
```

Example 3: List schedules - no criteria (POST)

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/wasscansched
ule" < file.xml
```

Note: “file.xml” contains the request POST data. Specify an empty file, since no search criteria is being specified.

Example 4: List active schedules - criteria (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/wasscansched
ule" < file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
    <filters>
    <Criteria field="active" operator="EQUALS">true</Criteria>
    <Criteria field="type"
operator="EQUALS">VULNERABILITY</Criteria>
    </filters>
</ServiceRequest>
```

Get schedule details

View details for a scheduled scan on a web application which is in the user's scope. Want to find a schedule ID to use as input? See Search schedules.

URL: https://qualysapi.qualys.com/qps/rest/3.0/get/was/wasscanschedule/<id>

Methods allowed: GET

Input

The element "id" (Integer) is required, where "id" identifies a scan.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Scan target must be within the user's scope

Example

Example 1: View schedule details (GET)

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/wasscanschedule
/1747"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>1747</id>
      <name><![CDATA[WEEKLY - Weekly Vulnerability Scan
Schedule]]></name>
      <owner>
```

```
    <id>123056</id>
    <username>username</username>
    <firstName><![CDATA[John]]></firstName>
    <lastName><![CDATA[Smith]]></lastName>
  </owner>
  <active>false</active>
  <type>VULNERABILITY</type>
  <target>
    <webApp>
      <id>324265</id>
      <name><![CDATA[Merchant Site]]></name>
    <url><![CDATA[http://10.10.25.116:80/merchant/2.2]]></url>
    </webApp>
    <scannerAppliance>
      <type>INTERNAL</type>
      <friendlyName><![CDATA[is_quays_js]]></friendlyName>
    </scannerAppliance>
  </target>
  <profile>
    <id>1072</id>
    <name><![CDATA[Initial WAS Options]]></name>
  </profile>
  <scheduling>
    <startDate>2014-08-12T18:30:00Z</startDate>
    <timeZone>
      <code>Europe/Paris</code>
      <offset>+01:00</offset>
    </timeZone>
    <occurrenceType>WEEKLY</occurrenceType>
    <cancelTime>11:00</cancelTime>
    <occurrence>
      <weeklyOccurrence>
        <everyNWeeks>2</everyNWeeks>
        <onDays>
          <WeekDay>MONDAY</WeekDay>
          <WeekDay>SATURDAY</WeekDay>
          <WeekDay>SUNDAY</WeekDay>
        </onDays>
        <occurrenceCount>20</occurrenceCount>
      </weeklyOccurrence>
    </occurrence>
```

```
        <cancelAfterNHours>11</cancelAfterNHours>
      </scheduling>
      <notification>
        <active>false</active>
      </notification>
      <launchedCount>0</launchedCount>
      <lastScan>
        <id>14930848885</id>
       <name><![CDATA[Web Application Vulnerability Scan]]></name>
        <reference>was/1399921142279.127704</reference>
        <launchedDate>2014-08-12T18:30:00Z</launchedDate>
        <status>FINISHED</status>
        <scanDuration>129</scanDuration>
      </lastScan>
      <createdDate>2014-08-12T18:00:00Z</createdDate>
      <createdBy>
        <id>123056</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </createdBy>
      <updatedDate>2014-08-12T18:00:00Z</updatedDate>
      <updatedBy>
        <id>123056</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
      <notification>
        <active>false</active>
        <reschedule>true</reschedule>
        <delay>
          <nb>1</nb>
          <scale>DAY</scale>
        </delay>
       <message><![CDATA[A QualysGuard scan is scheduled to start
soon.]]></message>
      </notification>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Create a schedule

Create a scheduled scan on a web application which is in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/create/was/wasscanschedule

Methods allowed: POST

Input

Required input elements are listed below. The associated data type for each element appears in parentheses, unless a compound element. See Reference: WasScanSchedule for descriptions of these <WasScanSchedule> elements.

Required Elements	Optional Elements
name (Text)	scannerAppliance
webApp.id (Integer)	webAppAuthRecord.id (Integer)
type (Keyword: DISCOVERY or VULNERABILITY)	options
profile.id (Integer)*	proxy.id (Integer)
startDate (Date)	cancelOption set to DEFAULT - Forces the use of the target web app's cancelScans option if set, else fall back to the one passed in to the API while launching the scan
timeZone (Text)	cancelOption set to SPECIFIC - Always use the cancel scan option passed while launching the scan
occurrenceType (Keyword: ONCE, DAILY, WEEKLY or MONTHLY)	
notification (Boolean)	
reschedule (Boolean)	

* The element profile (Text) is required unless the target has a default option profile.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Create WAS Schedule" permission
Scan target must be within the user's scope

Examples

Example 1: Create a new weekly schedule (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/wasscansched
ule" < file.xml
```

Request POST Data

```
<ServiceRequest>
 <data>
   <WasScanSchedule>
          <name><![CDATA[Create Schedule from API3 - using
Reschedule]]></name>
     <type>VULNERABILITY</type>
     <active>false</active>
     <scheduling>
       <cancelAfterNHours>8</cancelAfterNHours>
       <startDate>2014-09-06T09:50:11Z</startDate>
       <timeZone>
         <code>America/Vancouver</code>
        <offset>-07:00</offset>
       </timeZone>
       <occurrenceType>WEEKLY</occurrenceType>
       <occurrence>
          <weeklyOccurrence>
            <everyNWeeks>2</everyNWeeks>
            <occurrenceCount>20</occurrenceCount>
            <onDays>
              <WeekDay>SATURDAY</WeekDay>
              <WeekDay>SUNDAY</WeekDay>
            </onDays>
          </weeklyOccurrence>
       </occurrence>
     </scheduling>

    <notification>
        <active>true</active>
        <reschedule>true</reschedule>
```

```
        <delay>
          <nb>1</nb>
          <scale>DAY</scale>
        </delay>
       <message><![CDATA[A QualysGuard scan is scheduled to start
soon.]]></message>
      </notification>
     <target>
       <webApp>
             <id>1296335669</id>
</webApp>
        <webAppAuthRecord>
             <id>175535669</id>
</webAppAuthRecord>
     </target>
     <profile>
<id>712265669</id>
</profile>
   </WasScanSchedule>
 </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>203285669</id>
      <name><![CDATA[Create Schedule from API3 - using
Reschedule]]></name>
      <owner>
        <id>8792415669</id>
        <username>quays_cp</username>
        <firstName><![CDATA[Customer_2.6_1]]></firstName>
        <lastName><![CDATA[pocm]]></lastName>
      </owner>
```

```
      <active>false</active>
      <type>VULNERABILITY</type>
      <target>
        <webApp>
          <id>1296335669</id>
          <name><![CDATA[My Web Application]]></name>
          <url><![CDATA[http://10.10.26.238]]></url>
        </webApp>
        <webAppAuthRecord>
          <id>175535669</id>
          <name><![CDATA[AR1]]></name>
        </webAppAuthRecord>
        <scannerAppliance>
          <type>EXTERNAL</type>
        </scannerAppliance>
      </target>
      <profile>
        <id>712265669</id>
        <name><![CDATA[Initial WAS Options]]></name>
      </profile>
      <scheduling>
        <startDate>2014-09-06T09:50:00Z</startDate>
        <timeZone>
          <code>America/Vancouver</code>
          <offset>-07:00</offset>
        </timeZone>
        <occurrenceType>ONCE</occurrenceType>
        <cancelAfterNHours>8</cancelAfterNHours>
      </scheduling>
      <notification>
        <active>true</active>
        <reschedule>true</reschedule>
        <delay>
          <nb>1</nb>
          <scale>DAY</scale>
        </delay>
       <message><![CDATA[A QualysGuard scan is scheduled to start
soon.]]></message>
      </notification>
      <launchedCount>0</launchedCount>
      <createdDate>2014-08-27T22:30:59Z</createdDate>
```

```
      <createdBy>
        <id>8792415669</id>
        <username>quays_cp</username>
        <firstName><![CDATA[Customer_2.6_1]]></firstName>
        <lastName><![CDATA[pocm]]></lastName>
      </createdBy>
      <updatedDate>2014-08-27T22:31:00Z</updatedDate>
      <updatedBy>
        <id>8792415669</id>
        <username>acme_ab1</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Example 2: Create a new schedule - cancel scan option (POST)

Create a new vulnerability scan schedule on web app ID 2376281 and set the cancel scan option to SPECIFIC. Scans launched from this schedule will always use the cancel scan option passed with the schedule settings and will override the target web app's cancel scan setting, if set.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/wasscansched
ule" < file.xml
```

Request POST Data

```
ServiceRequest>
 <data>
   <WasScanSchedule>
     <name><![CDATA[My Scan Schedule]]></name>
     <type>VULNERABILITY</type>
     <scheduling>
        <cancelAfterNHours>7</cancelAfterNHours>
       <startDate>2015-09-30T13:11:00Z</startDate>
       <timeZone>
         <code>America/Dawson</code>
```

```
        </timeZone>
        <occurrenceType>ONCE</occurrenceType>
      </scheduling>
      <target>
        <webApp>
            <id>2376281</id>
        </webApp>
        <scannerAppliance>
          <type>EXTERNAL</type>
        </scannerAppliance>
        <cancelOption>SPECIFIC</cancelOption>
      </target>
      <profile>
         <id>332147</id>
      </profile>
    </WasScanSchedule>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>325624</id>
      <name><![CDATA[My Scan Schedule]]></name>
      <owner>
        <id>2086786</id>
        <username>acme_tp16</username>
        <firstName><![CDATA[FIRSTNAME]]></firstName>
        <lastName><![CDATA[LASTNAME]]></lastName>
      </owner>
      <active>true</active>
      <type>VULNERABILITY</type>
      <target>
        <webApp>
```

```
      <id>2376281</id>
      <name><![CDATA[My Web App]]></name>
      <url><![CDATA[http://10.10.26.238]]></url>
    </webApp>
    <scannerAppliance>
      <type>EXTERNAL</type>
    </scannerAppliance>
    <cancelOption>SPECIFIC</cancelOption>
  </target>
  <progressiveScanning>DEFAULT</progressiveScanning>
  <profile>
    <id>332147</id>
    <name><![CDATA[10 links]]></name>
  </profile>
  <scheduling>
    <startDate>2015-09-30T13:11:00Z</startDate>
    <timeZone>
      <code>America/Dawson</code>
      <offset>-07:00</offset>
    </timeZone>
    <occurrenceType>ONCE</occurrenceType>
    <cancelAfterNHours>7</cancelAfterNHours>
  </scheduling>
  <notification>
    <active>false</active>
  </notification>
  <nextLaunchDate>2015-09-30T20:11:00Z</nextLaunchDate>
  <launchedCount>0</launchedCount>
  <createdDate>2015-06-26T20:54:30Z</createdDate>
  <createdBy>
    <id>2086786</id>
    <username>ACME_tp16</username>
    <firstName><![CDATA[FIRSTNAME]]></firstName>
    <lastName><![CDATA[LASTNAME]]></lastName>
  </createdBy>
  <updatedDate>2015-06-26T20:54:30Z</updatedDate>
  <updatedBy>
    <id>2086786</id>
    <username>quays_tp16</username>
    <firstName><![CDATA[FIRSTNAME]]></firstName>
    <lastName><![CDATA[LASTNAME]]></lastName>
```

```
      </updatedBy>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Update a schedule

Update a scheduled scan on a web application which is in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/update/was/wasscanschedule/<id>

Methods allowed: POST

Input

The "id" (Integer) element and the data to be updated in the schedule are required where "id" identifies a schedule. See Reference: WasScanSchedule for descriptions of all of the <WasScanSchedule> elements.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Edit WAS Schedule" permission
Scan target must be within the user's scope

Example

Example 1: Enable notification (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/wasscanschedule/1688" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
    <WasScanSchedule>
      <notification>
        <active>true</active>
        <delay>
          <nb>4</nb>
          <scale>DAY</scale>
```

```
        </delay>
        <recipients>
          <set>
<EmailAddress><![CDATA[name1@company.com]]></EmailAddress>
<EmailAddress><![CDATA[name2@company.com]]></EmailAddress>
<EmailAddress><![CDATA[name3@company.com]]></EmailAddress>
          </set>
        </recipients>
        <message><![CDATA[The schedule notification
message]]></message>
      </notification>
    </WasScanSchedule>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>1688</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Example 2: Update notification to reschedule (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/wasscansched
ule/171425669” < file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
 <data>
   <WasScanSchedule>
<name><![CDATA[Update Notification to enable Reschedule]]></name>
 <notification>
        <active>true</active>
        <reschedule>true</reschedule>
        <delay>
          <nb>1</nb>
          <scale>DAY</scale>
        </delay>
       <message><![CDATA[A QualysGuard scan is scheduled to start
soon.]]></message>
      </notification>
   </WasScanSchedule>
 </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>171425669</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Activate an existing schedule

Activate one or more scheduled scans on web applications which are in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/update/was/wasscanschedule/<id>
or
https://qualysapi.qualys.com/qps/rest/3.0/activate/was/wasscanschedule/<filters>

Methods allowed: POST

Input

The "id" (Integer) element is required to activate a specific schedule where "id" identifies a schedule. Filters are required to activate schedules based on criteria. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: WasScanSchedule for descriptions of these <WasScanSchedule> elements.

id (Integer)	type (Keyword: VULNERABILITY or DISCOVERY)
name (Text)	active (Boolean)
webApp.id (Integer)	invalid (Boolean)
webApp.name (Text)	createdDate (Date)
owner.id (Integer)	updatedDate (Date)

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Edit WAS Schedule" permission
Scan target must be within the user's scope

Example

Activate a schedule (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/wasscansched
ule/1688" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
    <WasScanSchedule>
      <active>true</active>
    </WasScanSchedule>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>1688</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Deactivate an existing schedule

Deactivate one or more scheduled scans on web applications which are in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/update/was/wasscanschedule/<id>
or
https://qualysapi.qualys.com/qps/rest/3.0/deactivate/was/wasscanschedule/<filters>

Methods allowed: POST

Input

The "id" (Integer) element is required to deactivate a specific schedule where "id" identifies a schedule. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: WasScanSchedule for descriptions of these <WasScanSchedule> elements.

id (Integer)	type (Keyword: VULNERABILITY or DISCOVERY)
name (Text)	active (Boolean)
webApp.id (Integer)	invalid (Boolean)
webApp.name (Text)	createdDate (Date)
owner.id (Integer)	updatedDate (Date)

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Edit WAS Schedule" permission
Scan target must be within the user's scope

Example

Deactivate a schedule (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/wasscansched
ule/1688" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
    <WasScanSchedule>
      <active>false</active>
    </WasScanSchedule>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>1688</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Delete one or more existing schedules

Delete scheduled scans on web applications which are in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/delete/was/wasscanschedule/<id>
or
https://qualysapi.qualys.com/qps/rest/3.0/delete/was/wasscanschedule/<filters>

Methods allowed: POST

Input

The "id" (Integer) element is required to delete a specific schedule where "id" identifies a schedule. Filters are required to delete schedules based on criteria. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: WasScanSchedule for descriptions of these <WasScanSchedule> elements.

id (Integer)	type (Keyword: VULNERABILITY or DISCOVERY)
name (Text)	active (Boolean)
webApp.id (Integer)	invalid (Boolean)
webApp.name (Text)	createdDate (Date)
owner.id (Integer)	updatedDate (Date)

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Delete WAS Schedule" permission
Scan target must be within the user's scope

Examples

Example 1: Delete single schedule (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/wasscansched
ule/1846"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.cm/qps/xsd
/3.0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>1846</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Example 2: Delete schedules matching criteria (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/wasscansched
ule/" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
    <filters>
    <Criteria field="active" operator="EQUALS">false</Criteria>
<Criteria field="name" operator="CONTAINS">WEEKLY -</Criteria>
    </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
```

```
  <data>
    <WasScanSchedule>
      <id>1747</id>
    </WasScanSchedule>
    <WasScanSchedule>
      <id>1768</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Download one or more schedules to iCalendar

Download scheduled scans on a web applications, which are in the user's scope, to iCalendar format and then import them into your favorite calendar application so you can access your schedules on the go. You can import your schedules into several calendars including Microsoft Outlook, Google Calendar and Apple iCal.

URL: https://qualysapi.qualys.com/qps/rest/3.0/download/was/wasscanschedule/<id>
or
https://qualysapi.qualys.com/qps/rest/3.0/download/was/wasscanschedule/<filters>

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: WasScanSchedule for descriptions of these <WasScanSchedule> elements.

id (Integer)	type (Keyword: DISCOVERY or VULNERABILITY)
name (Text)	webApp.name (Text)
owner.id	webApp.id (Integer)
createdDate (Date)	updatedDate (Date)
active (Boolean)	invalid (Boolean)

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Scan target must be within the user's scope

Examples

Example 1: Download single (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/download/was/wasscansch
edule/1846”
```

Response:

```
BEGIN:VCALENDAR
PRODID:-//Qualys Inc//WAS Product//EN
VERSION:2.0
CALSCALE:GREGORIAN
METHOD:PUBLISH
BEGIN:VTIMEZONE
TZID:America/Boise
TZURL:http://tzurl.org/zoneinfo/America/Boise
X-LIC-LOCATION:America/Boise
BEGIN:DAYLIGHT
TZOFFSETFROM:-0700
TZOFFSETTO:-0600
TZNAME:MDT
DTSTART:20070311T020000
RRULE:FREQ=YEARLY;BYMONTH=3;BYDAY=2SU
END:DAYLIGHT
BEGIN:STANDARD
TZOFFSETFROM:-0600
TZOFFSETTO:-0700
TZNAME:MST
DTSTART:20071104T020000
RRULE:FREQ=YEARLY;BYMONTH=11;BYDAY=1SU
END:STANDARD
BEGIN:STANDARD
TZOFFSETFROM:-074449
TZOFFSETTO:-0800
TZNAME:PST
DTSTART:18831118T121511
END:STANDARD
BEGIN:DAYLIGHT
TZOFFSETFROM:-0800
```

```
TZOFFSETTO:-0700
TZNAME:PDT
DTSTART:19180331T020000
END:DAYLIGHT
BEGIN:STANDARD
TZOFFSETFROM:-0700
TZOFFSETTO:-0800
TZNAME:PST
DTSTART:19181027T020000
END:STANDARD
BEGIN:STANDARD
TZOFFSETFROM:-0800
TZOFFSETTO:-0700
TZNAME:MST
DTSTART:19230513T020000
END:STANDARD
BEGIN:DAYLIGHT
TZOFFSETFROM:-0700
TZOFFSETTO:-0600
TZNAME:MWT
DTSTART:19420209T020000
END:DAYLIGHT
BEGIN:DAYLIGHT
TZOFFSETFROM:-0600
TZOFFSETTO:-0600
TZNAME:MPT
DTSTART:19450814T170000
END:DAYLIGHT
BEGIN:STANDARD
TZOFFSETFROM:-0600
TZOFFSETTO:-0700
TZNAME:MST
DTSTART:19450930T020000
END:STANDARD
BEGIN:DAYLIGHT
TZOFFSETFROM:-0700
TZOFFSETTO:-0600
TZNAME:MDT
DTSTART:19670430T020000
END:DAYLIGHT
BEGIN:STANDARD
```

```
TZOFFSETFROM:-0700
TZOFFSETTO:-0700
TZNAME:MST
DTSTART:19740101T000000
END:STANDARD
END:VTIMEZONE

BEGIN:VEVENT
DTSTART;TZID=America/Boise:20111128T140000
DTEND;TZID=America/Boise:20111128T140000
RRULE:FREQ=DAILY;COUNT=1;INTERVAL=1
DTSTAMP:20120327T182146Z
UID:UID_ADDRESS
ORGANIZER:MAILTO:EMAIL_ADDRESS
CONTACT:John Smith (USERNAME)
CATEGORIES:Web Application Vulnerability Scan
LOCATION:my_scanner
DESCRIPTION:Type: Web Application Vulnerability Scan\n
 Web Application: My Apache\n
 Authentication Record: None\n
 Option Profile: My Apache Site - OP\n
 Owner: John Smith (USERNAME)\n
 Owner Phone: xxx-xxx-xxxx\n
 Owner Email: mailto:EMAIL_ADDRESS\n
 Modified By: John Smith (USERNAME)\n
 Time Zone: (GMT -06:00) Mountain Standard Time (MDT
America/Boise)
 Daylight Saving Time: On\n
 Duration: No run time threshold\n
 Approximate Duration: Not Available\n
 Occurs: Every 1 day\, End after 1 occurrence (1 remaining)
 Scanner Appliance: my_scanner\n
 Targets: My Web AppNone\n
 Notification: Off\n
SUMMARY:My Web App Scan Schedule (Active)
CREATED:20111128T204534Z
LAST-MODIFIED:20111128T210007Z
SEQUENCE:0
STATUS:CONFIRMED
TRANSP:TRANSPARENT
END:VEVENT
```

```
END:VCALENDAR
```

Example 2: Download with criteria (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/download/was/wasscansch
edule/" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
    <filters>
    <Criteria field="active" operator="EQUALS">false</Criteria>
<Criteria field="name" operator="CONTAINS">WEEKLY -</Criteria>
    </filters>
</ServiceRequest>
```

Reference: WasScanSchedule

The <WasScanSchedule> element includes sub elements used to define a web application scan schedule. A reference of these elements is provided below. An asterisk * indicates a complex element.

Element (data type)	Description
id (Integer)	The schedule ID. This element is assigned by the service and is required for a certain type of request (details, activate, deactivate).

owner*	The user who owns the schedule. User properties include user ID, user login, first and last name.
	Example:
	`<owner>`
	`  <id>123056</id>`
	`  <username>username</username>`
	`  <firstName><![CDATA[John]]></firstName>`
	`  <lastName><![CDATA[Smith]]></lastName>`
	`</owner>`

active (Boolean)	The schedule is active: true or false.
launchedCount (Integer)	The number of times the scan has been launched.
nextLaunchDate (Date)	The next launch date and time in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).

Element (data type)	Description
target*	The target of the scan. <webApp> is the target web application. <scannerAppliance> - type (Keyword) is set to INTERNAL for a scanner appliance, or EXTERNAL for external scanners. If the type is INTERNAL, friendlyName (Text) is the user-defined appliance name. <cancelOption> set to DEFAULT - Forces the use of the target web app's cancelScans option if set, else fall back to the one passed in to the API with the schedule settings. <cancelOption> set to SPECIFIC - Always use the cancel scan option passed with the schedule settings. Example: <target> <webApp> <id>324265</id> <name><![CDATA[Merchant Site]]></name> <url><![CDATA[http://url]]></url> </webApp> <scannerAppliance> <type>INTERNAL</type> <friendlyName><![CDATA[name]]></friendlyName> </scannerAppliance> <cancelOption>SPECIFIC</cancelOption> </target>
profile.id (Integer)	The name of the option profile that includes scan settings. The service provides the profile "Initial WAS Options" and we recommend this for getting started. Example: <profile> <name><![CDATA[Initial WAS Options]]></name> </profile>
proxy.id (Integer)	The proxy for scanning the target web application. Example: <proxy> <id>12345</id> </proxy>
createdDate (Date)	The schedule creation date and time in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).

Element (data type)	Description

createdBy*	The user who created the schedule.
	Example:
	`<createdBy>`
	`  <id>123056</id>`
	`  <username>username</username>`
	`  <firstName><![CDATA[John]]></firstName>`
	`  <lastName><![CDATA[Smith]]></lastName>`
	`</createdBy>`

updatedDate (Date)	The date and time of the most recent update of the schedule in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).

updatedBy*	The user who updated the schedule.
	Example:
	`<updatedBy>`
	`  <id>123056</id>`
	`  <username>username</username>`
	`  <firstName><![CDATA[John]]></firstName>`
	`  <lastName><![CDATA[Smith]]></lastName>`
	`</updatedBy>`

Element (data type)	Description

scheduling*	The schedule settings. <cancelAfterNHours> is the number of hours after which the scan task will be cancelled. <cancelTime> is the time at which a scan will be cancelled. <startDate> is the date and time the scan will begin. <timeZone> is the time zone that applies to the schedule. <occurence> defines frequency of the task: SINGLE, DAILY, WEEKLY or MONTHLY.
	Example of weekly scan with the <cancelAfterNHours> option:
	`<scheduling>`
	`  <cancelAfterNHours>11</cancelAfterNHours>`
	`  <startDate>2012-02-02T10:10:00Z</startDate>`
	`   <timeZone>`
	`     <code>Europe/Paris</code>`
	`   </timeZone>`
	`   <occurrenceType>WEEKLY</occurrenceType>`
	`  <occurrence>`
	`    <weeklyOccurrence>`
	`      <everyNWeeks>2</everyNWeeks>`
	`       <occurrenceCount>20</occurrenceCount>`
	`       <onDays>`
	`        <WeekDay>MONDAY</WeekDay>`
	`        <WeekDay>SATURDAY</WeekDay>`
	`         <WeekDay>SUNDAY</WeekDay>`
	`      </onDays>`
	`    </weeklyOccurrence>`
	`  </occurrence>`
	`</scheduling>`
	
	Example of single occurrence scan with the <cancelTime> optton:
	`<scheduling>`
	`   <startDate>2012-02-02T10:10:00Z</startDate>`
	`   <timeZone>`
	`     <code>Europe/Paris</code>`
	`   </timeZone>`
	`   <occurrenceType>ONCE</occurrenceType>`
	`   <occurrence>`
	`    <cancelTime>11:15</cancelTime>`
	`  </occurrence>`
	`</scheduling>`

Element (data type)	Description

notification*	The notification settings. <active> indicates whether notification is enabled. <delay> indicates when the notification will be sent as number of days, hours, or minutes before the scan. <scale> indicates the delay unit: DAY, HOUR or MINUTE. <recipients> identifies the email addresses of the notification recipients. <message> is the text of the notification message.
	Example:
	`<notification>`
	`  <active>true</active>`
	`  <delay>`
	`  <nb>1</nb>`
	`  <scale>DAY</scale>`
	`  </delay>`
	`  <recipients>`
	`  <set>`
	`<EmailAddress><![CDATA[1@a.com]]></EmailAddress>`
	`<EmailAddress><![CDATA[2@a.com]]></EmailAddress>`
	`  </set>`
	`  </recipients>`
	`  <message><![CDATA[The message]]></message>`
	`</notification>`

CHAPTER 6

Report API

The WAS Report API provides a suite of API functions for managing web application reports across the enterprise.

These operations are available:

Current report count

Search reports

Get report details

Get report status

Download a report

Send an encrypted PDF report

Update a report

Delete one or more existing reports

Current report count

Returns the total number of reports in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/count/was/report

Methods allowed: GET, POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: Report for descriptions of these <Report> elements.

id (Integer)	creationDate (Date)
name (Text)	type (Keyword: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT or DATALIST_REPORT)
tags.id (Integer)	format (Keyword: HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, CSV, XML, POWERPOINT or WORD)
tags.name (Text)	status (Keyword: RUNNING, ERROR or COMPLETE)

Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Count includes reports within the user's scope

Examples

Example 1: Count - no criteria (GET)

Request:

```
curl -u “USERNAME:PASSWORD”
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/report"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse>
  <count>12</count>
  <responseCode>SUCCESS</responseCode>
</ServiceResponse>
```

Example 2: Count - criteria (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X “POST”
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/report" <
file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
    <filters>
        <Criteria field="id" operator="IN">1302, 1303</Criteria>
    </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse>
  <count>1</count>
  <responseCode>SUCCESS</responseCode>
</ServiceResponse>
```

Search reports

Returns a list of reports which are in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/search/was/report

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: Report for descriptions of these <Report> elements.

id (Integer)	creationDate (Date)
name (Text)	type (Keyword: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT or DATALIST_REPORT)
tags	format (Keyword: HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, CSV, XML, POWERPOINT or WORD)
tags.id (Integer)	status (Keyword: RUNNING, ERROR or COMPLETE)
tags.name (Text)	

Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Output includes reports within the user's scope

Examples

Example 1: Search - no criteria (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/report" <
file.xml
```

Note: "file.xml" contains the request POST data. Specify an empty file, since no search criteria is being specified.

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse>
  <count>3</count>
  <data>
    <list>
      <Report>
        <id>1393</id>
        <name><![CDATA[Web Application Report 1]]></name>
        <type>WAS_WEBAPP_REPORT</type>
        <format>PDF</format>
        <status>COMPLETE</status>
        <size>2244667</size>
        <creationDate>2011-11-25T10:20:06Z</creationDate>
        <tags>
          <count>0</count>
        </tags>
        <owner>
          <id>123056</id>
          <username>username</username>
          <firstName><![CDATA[John]]></firstName>
          <lastName><![CDATA[Smith]]></lastName>
        </owner>
      </Report>
      <Report>
        <id>1394</id>
        <name><![CDATA[Web Application Report 2]]></name>
        <type>WAS_WEBAPP_REPORT</type>
        <format>PDF</format>
```

```
            <status>COMPLETE</status>
            <size>124578</size>
            <creationDate>2011-11-25T10:21:25Z</creationDate>
            <tags>
              <count>0</count>
            </tags>
            <owner>
              <id>123056</id>
              <username>username</username>
              <firstName><![CDATA[John]]></firstName>
              <lastName><![CDATA[Smith]]></lastName>
            </owner>
          </Report>
          <Report>
            <id>1282</id>
            <name><![CDATA[Web Application Report 3]]></name>
            <type>WAS_WEBAPP_REPORT</type>
            <format>PDF</format>
            <status>COMPLETE</status>
            <size>12341234</size>
            <creationDate>2011-11-24T00:00:00Z</creationDate>
            <tags>
              <count>0</count>
            </tags>
            <owner>
              <id>123056</id>
              <username>username</username>
              <firstName><![CDATA[John]]></firstName>
              <lastName><![CDATA[Smith]]></lastName>
            </owner>
          </Report>
        </list>
      </data>
      <isDone>true</isDone>
      <responseCode>SUCCESS</responseCode>
      <responseErrorDetails>
        <internalErrorCodeId>0</internalErrorCodeId>
      </responseErrorDetails>
    </ServiceResponse>
```

Example 2: Search - criteria (POST)

Request:

```
curl -u “USERNAME:PASSWORD” -H “content-type: text/xml” -X “POST”
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/report" <
file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
    <filters>
        <Criteria field="tags.id"
operator="EQUALS">99511</Criteria>
    </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse>
  <count>1</count>
  <data>
    <list>
      <Report>
        <id>1302</id>
        <name><![CDATA[Web Application Report 2]]></name>
        <type>WAS_WEBAPP_REPORT</type>
        <format>PDF_ENCRYPTED</format>
        <status>COMPLETE</status>
        <size>2244667</size>
        <creationDate>2011-11-24T00:00:00Z</creationDate>
        <tags>
          <count>1</count>
        </tags>
        <distributionList>
          <count>12</count>
        </distributionList>
        <owner>
          <id>123056</id>
          <username>username</username>
          <firstName><![CDATA[John]]></firstName>
```

```
            <lastName><![CDATA[Smith]]></lastName>
          </owner>
        </Report>
      </list>
    </data>
    <isDone>true</isDone>
    <responseCode>SUCCESS</responseCode>
    <responseErrorDetails>
      <internalErrorCodeId>0</internalErrorCodeId>
    </responseErrorDetails>
  </ServiceResponse>
```

Get report details

View details for a report which is in the user's scope. Want to find a report ID to use as input? See Search reports.

URL: https://qualysapi.qualys.com/qps/rest/3.0/get/was/report/<id>

Methods allowed: GET

Input

The element "id" (Integer) is required, where id identifies a report.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Report must be within the user's scope

Example

Get report details

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/report/1302"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse>
  <count>1</count>
  <data>
    <Report>
      <id>1302</id>
      <name><![CDATA[Web Application Report 2]]></name>
      <type>WAS_WEBAPP_REPORT</type>
      <format>PDF_ENCRYPTED</format>
      <status>COMPLETE</status>
      <size>2244667</size>
      <creationDate>2011-11-24T00:00:00Z</creationDate>
      <lastDownloadDate>2011-11-09T00:00:00Z</lastDownloadDate>
      <downloadCount>1</downloadCount>
      <tags>
```

```
          <count>2</count>
          <list>
            <Tag>
              <id>99509</id>
              <name><![CDATA[Tag 1]]></name>
            </Tag>
            <Tag>
              <id>99510</id>
              <name><![CDATA[Tag 2]]></name>
            </Tag>
          </list>
        </tags>
        <distributionList>
          <count>2</count>
          <list>

<EmailAddress><![CDATA[email1@company.com]]></EmailAddress>

<EmailAddress><![CDATA[email2@company.com]]></EmailAddress>
          </list>
        </distributionList>
        <owner>
          <id>123056</id>
          <username>username</username>
          <firstName><![CDATA[John]]></firstName>
          <lastName><![CDATA[Smith]]></lastName>
        </owner>
      </Report>
    </data>
    <responseCode>SUCCESS</responseCode>
</ServiceResponse>
```

Get report status

Retrieve the status of a report which is in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/status/was/report/<id>

Methods allowed: GET

Input

The element "id" (Integer) is required, where "id" identifies a report.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Report must be within the user's scope

Example

Get report status

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/status/was/report/1302"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse>
  <count>1</count>
  <data>
    <Report>
      <id>1302</id>
      <status>COMPLETE</status>
    </Report>
  </data>
  <responseCode>SUCCESS</responseCode>
</ServiceResponse>
```

Download a report

Download a report which is in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/download/was/report/<id>

Methods allowed: GET

Input

The elements "id" (Integer) is required, where "id" identifies a report.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Report must be within the user's scope

Example

Get report status

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/download/was/report/130
2"
```

Response:

Report ID 1302 will be downloaded in the format in which it was generated.

Send an encrypted PDF report

Send an encrypted PDF report, which is in the user's scope, to a distribution list.

URL: https://qualysapi.qualys.com/qps/rest/3.0/send/was/report/<id>

Methods allowed: POST

Input

The elements "id" (Integer) and "distributionList" (Text) are required, where "id" identifies a report and "distributionList" identifies the email addresses of the report recipients.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Distribute Report" permission
Report must be within the user's scope

Example

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/send/was/report/1302" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
    <data>
        <Report>
            <distributionList>
                <add>

<EmailAddress><![CDATA[email1@abc.com]]></EmailAddress>

<EmailAddress><![CDATA[email2@abc.com]]></EmailAddress>
                </add>
            </distributionList>
```

```
        </Report>
    </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse>
  <count>1</count>
  <data>
    <Report>
      <id>1302</id>
    </Report>
  </data>
  <responseCode>SUCCESS</responseCode>
</ServiceResponse>
```

Update a report

Update the tags assigned to a report which is in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/update/was/report/<id>

Methods allowed: POST

Input

The elements "id" (Integer) and "tags" (complex element) are required, where "id" identifies a report and "tags" identifies tags to be added or removed.

The element "showPatched" can be set to filter the report to include/not include findings with virtual patches. Applies to Web Application Report and Scan Report. This filter can be set to:

SHOW_ONLY - show patched findings only
SHOW_BOTH - show patched & unpatched findings (default)
SHOW_NONE - show unpatched findings only

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Edit Report" permission
Report must be within the user's scope

Example

Update a report - add a tag

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/report/1304"
< file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
    <data>
        <Report>
```

```
            <tags>
                <set>
                    <Tag>
                        <id>99509</id>
                    </Tag>
                    <Tag>
                        <id>99510</id>
                    </Tag>
                </set>
            </tags>
        </Report>
    </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse>
  <count>1</count>
  <data>
    <Report>
      <id>1304</id>
    </Report>
  </data>
  <responseCode>SUCCESS</responseCode>
</ServiceResponse>
```

Delete one or more existing reports

Delete a report which is in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/delete/was/report/<id>
or
https://qualysapi.qualys.com/qps/rest/3.0/delete/was/report

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. To delete one report by the report ID, the id element is required. the other elements listed below are used to delete reports based on filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: Report for descriptions of these <Report> elements.

id (Integer)	creationDate (Date)
name (Text)	type (Keyword: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT or DATALIST_REPORT)
tags.id (Integer)	format (Keyword: HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, CSV, XML, POWERPOINT or WORD)
tags.name (Text)	status (Keyword: RUNNING, ERROR or COMPLETE)

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Delete Report" permission
Report must be within the user's scope

Examples

Example 1: Delete a single report (POST)

Delete the report with the ID 1393.

Request:

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/report/6333"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/report.xsd">
 <responseCode>SUCCESS</responseCode>
 <count>1</count>
 <data>
   <Report>
     <id>6333</id>
   </Report>
 </data>
```

Example 2: Delete reports - criteria (POST)

Delete reports matching one or both of these criteria: 1) reports with names that contain the string "to be deleted", and 2) reports that are completed (having the status COMPLETED).

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/report" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">to be
deleted</Criteria>
   <Criteria field="status" operator="EQUALS">COMPLETE</Criteria>
```

```
  </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/report.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Report>
      <id>1542</id>
    </Report>
  </data>
</ServiceResponse>
```

Reference: Report

The <Report> element includes sub elements used to define a web application report. A reference of these elements is provided below. An asterisk * indicates a complex element.

Element (data type)	Description
id (Integer)	The report ID. This element is assigned by the service and is required for a certain type of request (details, status, update, delete, send or download).
name (Text)	The report name (maximum 256 characters).
description (Text)	A description of the report.

owner*	This element is assigned by the service and may be specified for an update request only.
	Example:
	`<owner>`
	`  <id>123056</id>`
	`  <username>username</username>`
	`  <firstName><![CDATA[Johns]]></firstName>`
	`  <lastName><![CDATA[Smith]]></lastName>`
	`</owner>`

type (Text)	The report type, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT, DATALIST_REPORT
format (Text)	The format of the report, one of: HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, POWERPOINT, CSV, XML, WORD

tags*	This element identifies the tags associated with the report.
	Example:
	`<tags>`
	`  <count>2</count>`
	`  <list>`
	`  <Tag>`
	`  <id>99509</id>`
	`  <name><![CDATA[Tag 1]]></name>`
	`  </Tag>`
	`  <Tag>`
	`  <id>99511</id>`
	`  <name><![CDATA[Tag 2]]></name>`
	`  </Tag>`
	`  </list>`
	`</tags>`

password (Text)	The password for a PDF encrypted report.

Element (data type)	Description

distributionList*	This element specifies the email addresses for distribution of the report.
	Example:
	`<distributionList>`
	`  <count>2</count>`
	`  <list>`
	`<EmailAddress><![CDATA[1@abc.com]]></EmailAddress>`
	`<EmailAddress><![CDATA[2@abc.com]]></EmailAddress>`
	`  </list>`
	`</distributionList>`

config*	The configuration options for report creation.
	Example:
	`<config>`
	`  <webAppReport>`
	`  <target>`
	`  <tags>`
	`  <Tag>`
	`  <id>102609</id>`
	`  </Tag>`
	`  </tags>`
	`  <webapps>`
	`  <WebApp>`
	`  <id>324538</id>`
	`  </WebApp>`
	`  </webapps>`
	`  </target>`
	`...`

status (Keyword)	The status of the report: RUNNING, ERROR or COMPLETE
creationDate (Date)	The date when the report was created in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).
lastDownloadDate (Date)	The date when the report was last downloaded in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).
downloadCount (Integer)	The number of times the report has been downloaded.

CHAPTER 7

Report Creation API

The WAS Report Creation API allows you to create a variety of web application reports based on security information collected by the most recent scans of your web applications.

See these topics:

Report Creation API

Web Application Report

Scan Report

Scorecard Report

Catalog Report

Report Creation API

Using the Report Creation API you can create these reports: Web Application Report, Scan Report, Scorecard Report and Catalog Report.

URL: https://qualysapi.qualys.com/qps/rest/3.0/create/was/report

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. See Reference: Report Creation for descriptions of these elements.

Required Elements	Optional Elements
name (Text)	tags.id (Integer)
type (Keyword: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT or WAS_CATALOG_REPORT)	tags.name (Text)
format (Keyword: HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, CSV, XML, or POWERPOINT)	password (Text)
config*(1)	distributionList*

(1) The “config” element must have one and only one of these child elements: webAppReport, scanReport, catalogReport or scorecardReport. These are described in the next sections.

Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have "Create Report" permission

Web Application Report

A Web Application Report shows the security status of your web applications.

Input for this report

Allowed input elements are listed below. The element "target" is required and at least one of these child elements must be set: "tags" or "webapps". See Reference: Report Creation for descriptions of these elements.

target.tags (Tag)	display.contents (WebAppReportContent)
target.webapps (WebApp)	display.graphs (WebAppReportGraph)
filters.searchlists (SearchList)	display.groups (WebAppReportGroup)
filters.url (Text)	display.options (rawLevels)
filters.status (WebAppFindingStatus)	display.groups (WebAppReportGroup)
filters.remediation showPatched (Keyword: SHOW_ONLY, SHOW_NONE, SHOW_BOTH - default)	

Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN

Example

Create a web application report

Create a web application report in encrypted PDF format, setting both tags and web applications for the target.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/report" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
    <Report>
      <name><![CDATA[API Web Application Report]]></name>
      <description><![CDATA[Encrypted PDF WebApp
report]]></description>
      <format>PDF_ENCRYPTED</format>
      <password>PASSWORD</password>
      <distributionList>
        <set>
          <EmailAddress>email@domain.com</EmailAddress>
        </set>
      </distributionList>
      <type>WAS_WEBAPP_REPORT</type>
      <config>
        <webAppReport>
          <target>
            <tags>
              <Tag>
                <id>7821676</id>
              </Tag>
            </tags>
            <webapps>
              <WebApp>
                <id>28510007</id>
              </WebApp>
            </webapps>
```

```
            </target>
            <display>
              <contents>
              <WebAppReportContent>DESCRIPTION</WebAppReportContent>
                <WebAppReportContent>SUMMARY</WebAppReportContent>
                <WebAppReportContent>GRAPHS</WebAppReportContent>
                <WebAppReportContent>RESULTS</WebAppReportContent>
              </contents>
              <graphs>
<WebAppReportGraph>VULNERABILITIES_BY_GROUP</WebAppReportGraph>
<WebAppReportGraph>VULNERABILITIES_BY_OWASP</WebAppReportGraph>
              </graphs>
              <groups>
                <WebAppReportGroup>GROUP</WebAppReportGroup>
                <WebAppReportGroup>OWASP</WebAppReportGroup>
              </groups>
              <options>
                <rawLevels>true</rawLevels>
              </options>
            </display>
            <filters>
              <includedSearchLists>
                <SearchList>
                  <id>378688</id>
                </SearchList>
              </includedSearchLists>
              <status>
                <WebAppFindingStatus>NEW</WebAppFindingStatus>
                <WebAppFindingStatus>ACTIVE</WebAppFindingStatus>
                <WebAppFindingStatus>REOPENED</WebAppFindingStatus>
              </status>
            </filters>
          </webAppReport>
        </config>
      </Report>
    </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/report.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Report>
      <id>2629</id>
    </Report>
  </data>
</ServiceResponse>
```

Scan Report

A scan report shows you the results of scans on a particular web application.

Input for this report

Allowed input elements are listed below. The element "target" is required and at least one "scans" child element is required. See Reference: Report Creation for details.

target.scans (WasScan)	display.contents (ScanAppReportContent)
filters.searchlists (SearchList)	display.graphs (ScanAppReportGraph)
filters.url (Text)	display.groups (ScanAppReportGroup)
filters.status (ScanFindingStatus)	display.options (rawLevels)
filters.remediation showPatched (Keyword: SHOW_ONLY, SHOW_NONE, SHOW_BOTH - default)	

Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN

Example

Create a scan report

Create a scan report in HTML ZIPPED format, selecting a single scan for the target.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/report" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
    <Report>
      <name><![CDATA[with all parameters HTML_ZIPPED]]></name>
      <description><![CDATA[A simple scan report]]></description>
      <format>HTML_ZIPPED</format>
      <type>WAS_SCAN_REPORT</type>
      <config>
        <scanReport>
          <target>
            <scans>
              <WasScan>
                <id>104268</id>
              </WasScan>
            </scans>
          </target>
          <display>
            <contents>
              <ScanReportContent>DESCRIPTION</ScanReportContent>
              <ScanReportContent>SUMMARY</ScanReportContent>
              <ScanReportContent>GRAPHS</ScanReportContent>
              <ScanReportContent>RESULTS</ScanReportContent>
         <ScanReportContent>INDIVIDUAL_RECORDS</ScanReportContent>
             <ScanReportContent>RECORD_DETAILS</ScanReportContent>
              <ScanReportContent>ALL_RESULTS</ScanReportContent>
              <ScanReportContent>APPENDIX</ScanReportContent>
            </contents>
            <graphs>

<ScanReportGraph>VULNERABILITIES_BY_SEVERITY</ScanReportGraph>

<ScanReportGraph>VULNERABILITIES_BY_GROUP</ScanReportGraph>

<ScanReportGraph>VULNERABILITIES_BY_OWASP</ScanReportGraph>

<ScanReportGraph>VULNERABILITIES_BY_WASC</ScanReportGraph>

<ScanReportGraph>SENSITIVE_CONTENTS_BY_GROUP</ScanReportGraph>
            </graphs>
            <groups>
```

```
            <ScanReportGroup>URL</ScanReportGroup>
            <ScanReportGroup>GROUP</ScanReportGroup>
            <ScanReportGroup>OWASP</ScanReportGroup>
            <ScanReportGroup>WASC</ScanReportGroup>
            <ScanReportGroup>STATUS</ScanReportGroup>
            <ScanReportGroup>CATEGORY</ScanReportGroup>
            <ScanReportGroup>QID</ScanReportGroup>
          </groups>
          <options>
            <rawLevels>true</rawLevels>
          </options>
        </display>
        <filters>
          <searchlists>
            <SearchList>
              <id>43147</id>
            </SearchList>
          </searchlists>
          <url>http://www.mysite.com/help.html</url>
          <status>
            <ScanFindingStatus>NEW</ScanFindingStatus>
            <ScanFindingStatus>ACTIVE</ScanFindingStatus>
            <ScanFindingStatus>REOPENED</ScanFindingStatus>
            <ScanFindingStatus>FIXED</ScanFindingStatus>
          </status>
        </filters>
      </scanReport>
    </config>
  </Report>
 </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/report.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
```

```
    <Report>
      <id>3629</id>
    </Report>
  </data>
</ServiceResponse>
```

Scorecard Report

A Scorecard Report ranks the vulnerability of your web applications.

Input for this report

Allowed input elements are listed below. See Reference: Report Creation for descriptions of these elements.

target.tags (Tag)	display.contents (ScorecardReportContent)
filters.searchlists (SearchList)	display.graphs (ScorecardReportGraph)
filters.scanDate (DatetimeRange)	display.groups (ScorecardReportGroup)
filters.scanStatus (WasScanConsolidatedStatus)	display.options (rawLevels)
filters.scanAuthStatus (WasScanAuthStatus)	

Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN

Example

Create a scorecard report

Create a scorecard report in PDF format, selecting a single tag for the target.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/report" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
    <Report>
      <name><![CDATA[with all parameters PDF with rawLevel
false]]></name>
      <description><![CDATA[A simple scorecard
report]]></description>
      <format>PDF</format>
      <type>WAS_SCORECARD_REPORT</type>
      <config>
        <scorecardReport>
          <target>
            <tags>
              <Tag>
                <id>243130</id>
              </Tag>
            </tags>
          </target>
          <display>
            <contents>

<ScorecardReportContent>DESCRIPTION</ScorecardReportContent>

<ScorecardReportContent>SUMMARY</ScorecardReportContent>

<ScorecardReportContent>GRAPHS</ScorecardReportContent>

<ScorecardReportContent>RESULTS</ScorecardReportContent>
```

```
            </contents>
            <graphs>

<ScorecardReportGraph>VULNERABILITIES_BY_GROUP</ScorecardReportGra
ph>

<ScorecardReportGraph>VULNERABILITIES_BY_OWASP</ScorecardReportGra
ph>

<ScorecardReportGraph>VULNERABILITIES_BY_WASC</ScorecardReportGrap
h>
          </graphs>
          <groups>
            <scorecardReportGroup>GROUP</ScorecardReportGroup>
            <ScorecardReportGroup>OWASP</ScorecardReportGroup>
            <ScorecardReportGroup>WASC</ScorecardReportGroup>
          </groups>
          <options>
            <rawLevels>false</rawLevels>
          </options>
           </display>
           <filters>
             <searchlists>
               <SearchList>
                 <id>43147</id>
               </SearchList>
               <SearchList>
                 <id>43147</id>
               </SearchList>
             </searchlists>
             <scanDate>
               <startDate>2012-08-28</startDate>
               <endDate>2012-10-28</endDate>
             </scanDate>
             <scanStatus>NO_HOST_ALIVE</scanStatus>
             <scanAuthStatus>NONE</scanAuthStatus>
           </filters>
         </scorecardReport>
       </config>
     </Report>
   </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/report.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Report>
      <id>4629</id>
    </Report>
  </data>
</ServiceResponse>
```

Catalog Report

A Catalog Report shows you the number and status of entries in your web application catalog.

Input for this report

Allowed input elements are listed below. See Reference: Report Creation for descriptions of these elements.

filters.scanDate (DatetimeRange)	display.contents (WebAppReportContent)
filters.url (Text)	display.graphs (WebAppReportGraph)
filters.ip (Text)	display.groups (WebAppReportGroup)
filters.os (Text)	display.options (rawLevels)
filters.status (EntryStatus)	

Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN

Example

Create a catalog report

Create a catalog report in CSV format, selecting a single tag for the target.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/report" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
  <data>
    <Report>
      <name><![CDATA[with all parameters CSV]]></name>
      <description><![CDATA[A simple Catalog
report]]></description>
      <type>WAS_CATALOG_REPORT</type>
      <format>CSV</format>
      <config>
        <catalogReport>
          <display>
            <contents>

<CatalogReportContent>DESCRIPTION</CatalogReportContent>
              <CatalogReportContent>SUMMARY</CatalogReportContent>
              <CatalogReportContent>GRAPHS</CatalogReportContent>
              <CatalogReportContent>RESULTS</CatalogReportContent>

<CatalogReportContent>INDIVIDUAL_RECORDS</CatalogReportContent>
            </contents>
            <graphs>

<CatalogReportGraph>ENTRIES_ADDED_OVER_TIME</CatalogReportGraph>

<CatalogReportGraph>ENTRIES_BY_STATUS</CatalogReportGraph>
            </graphs>
            <groups>
                <CatalogReportGroup>STATUS</CatalogReportGroup>
```

```
<CatalogReportGroup>OPERATING_SYSTEM</CatalogReportGroup>
            </groups>
          </display>
          <filters>
            <status>
                <EntryStatus>NEW</EntryStatus>
              <EntryStatus>SUBSCRIPTION</EntryStatus>
              <EntryStatus>ROGUE</EntryStatus>
              <EntryStatus>APPROVED</EntryStatus>
              <EntryStatus>REJECTED</EntryStatus>
            </status>
            <scanDate>
              <startDate>2012-08-28T08:53:43Z</startDate>
              <endDate>2012-10-28T08:53:43Z</endDate>
            </scanDate>
            <url><![CDATA[mysite.fr]]></url>
            <os><![CDATA[unix]]></os>
          </filters>
        </catalogReport>
      </config>
    </Report>
  </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/report.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Report>
      <id>5629</id>
    </Report>
  </data>
</ServiceResponse>
```

Reference: Report Creation

The Report "config" element includes sub elements used to define a web application report type. A reference of these elements is provided below. An asterisk * indicates a complex element.

Element (data type)	Description
name (Text)	A report name (maximum 256 characters). Applies to all reports.

target*	A report target. Applies to all reports.
	Example for a web application report:
	`<target>`
	`  <tags>`
	`  <tag>`
	`  <id>1234</id>`
	`  </tag>`
	`  </tags>`
	`  <webapps>`
	`  <WebApp>`
	`  <id>2345</id>`
	`  </WebApp>`
	`  </webapps>`
	`</target>`

password (Text)	A password for a encrypted PDF report. Applies to all reports.

distributionList*	Email addresses for a report distribution list. Applies to all reports.
	Example:
	`<distributionList>`
	`  <set>`
	`  <EmailAddress><EMAIL_ADDRESS1></EmailAddress>`
	`  <EmailAddress><EMAIL_ADDRESS2></EmailAddress>`
	`  </set>`
	`</distributionList>`

display.contents*	Identifies the report content to display.
	Values: DESCRIPTION, SUMMARY, GRAPHS, RESULTS, INDIVIDUAL_RECORDS (all reports)
	Values: RECORD_DETAILS, ALL_RESULTS, APPENDIX (Web Application Report and Scan Report)
	Values: RESULTS_PAYLOADS (Scan Report)
	Example for a Scan Report:
	`<display>`
	`  <contents>`
	`  <ScanReportContent>GRAPHS</ScanReportContent>`
	`  <ScanReportContent>RESULTS</ScanReportContent>`
	`  </contents>`
	`</display>`

Element (data type)	Description
display.graphs*	Identifies the graphs to display. Applies to all reports. Example for a Scan Report: <display> <graphs> <ScanReportGraph>MOST_VULNERABLE_URLS</ScanReportGraph> <ScanReportGraph>VULNERABILITIES_BY_SEVERITY</ScanReportGraph> <ScanReportGraph>VULNERABILITIES_BY_GROUP</ScanReportGraph> <ScanReportGraph>VULNERABILITIES_BY_OWASP</ScanReportGraph> <ScanReportGraph>VULNERABILITIES_BY_WASC</ScanReportGraph> <ScanReportGraph>SENSITIVE_CONTENTS_BY_GROUP</ScanReportGraph> </graphs> </display>
display.groups*	Identifies the vulnerability groups to display. Applies to all reports. Example for a Web Application Report or Scan Report: <display> <groups> <WebAppReportGroup>GROUP</WebAppReportGroup> <WebAppReportGroup>OWASP</WebAppReportGroup> <WebAppReportGroup>WASC</WebAppReportGroup> </groups> </display>
display.options*	Specifies whether to display severity using levels (1 through 5) or using ratings (low, medium, high). Applies to all reports.
filters.searchlists*	Identifies search list filters. Applies to a Web Application Report, Scan Report or Scorecard Report. Example: <filters> <SearchlLsts> <SearchList> <id>43147</id> </SearchList> </SearchlLsts> </filters>

Element (data type)	Description
filters.url (Text)	Identifies URL filters. Applies to a Web Application Report, Scan Report or Catalog Report. Example: <filters> <url>http://www.mysite.com/help.html</url> ... </filters>
filters.status*	Identifies status filters. Applies to Web Application Report, Scan Report and Catalog Report. Values for Web Application Report and Scan Report: NEW, ACTIVE, REOPENED, FIXED, IGNORED (Web Application Report only) Values for Catalog Report: NEW, ROGUE, APPROVED, REJECTED, SUBSCRIPTION
filters.showPatched (Keyword)	Identifies whether to include/not include findings with virtual patches. Applies to Web Application Report and Scan Report. Values: SHOW_ONLY - show patched findings only SHOW_BOTH - show patched & unpatched findings (default) SHOW_NONE - show unpatched findings only
filters.scanDate*	Applies to a Scorecard Report and Catalog Report: Example: <filters> <scanDate> <startDate>2012-08-28</startDate> <endDate>2012-10-28</endDate> </scanDate> </filters>
filters.scanStatus*	Applies to a Scorecard Report (don't see values in the structure) Example: <filters> <scanStatus>FINISHED</scanStatus> </filters>
filters.scanAuthStatus*	Applies to a Scorecard Report (don't see values in the structure) Example: <filters> <scanAuthStatus>SUCCESSFUL</scanAuthStatus> </filters>

Element (data type)	Description

filters.ip (Text)	Applies to a Catalog Report
	Example:
	`<filters>`
	`<ip><![CDATA[10.56.64.245]]></ip>`
	`</filters>`

filters.os (Text)	Applies to a Catalog Report
	Example:
	`<filters>`
	`<os><![CDATA[unix]]></os>`
	`</filters>`

CHAPTER 8

Option Profile API

The WAS Option Profile API provides a suite of API functions for managing option profiles you want to use for scanning web applications.

These operations are available:

Current option profile count

Search option profiles

Get details for an option profile

Create a new option profile

Update an option profile

Delete an option profile

Current option profile count

Returns the total number of option profiles in the user's scope. Input elements are optional and are used to filter the number of option profiles included in the count.

URL: https://qualysapi.qualys.com/qps/rest/3.0/count/was/option profile

Methods allowed: GET, POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

id (Integer)	Filter by option profile ID
name (Text)	Filter by option profile name
tags	Filter by tags applied
tags.id (Integer)	Filter by tag ID applied
tags.name (Text)	Filter by tag name applied
createdDate (Date)	Filter by created date (UTC date/time format)
updatedDate (Date)	Filter by last updated date (UTC date/time format)
usedByWebApps (Boolean with operator: EQUALS or NOT EQUALS)	Filter profiles if used/not used by web applications
usedBySchedules (Boolean with operator: EQUALS or NOT EQUALS)	Filter profiles if used/not used by scan schedules
owner.id (Long with operator: EQUALS, IN, NOT EQUALS, GREATER or LESSER)	Filter profiles based on owner's user ID
owner.name (text with operator: CONTAINS, EQUALS or NOT EQUALS)	Filter profiles based on owner's full name (first and last)
owner.username (text with operator: CONTAINS, EQUALS or NOT EQUALS)	Filter profiles based on owner's username (like acme_ab3)

Allowed Operators	
Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN
Boolean	(true/false) EQUALS, NOT EQUALS

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Count includes option profiles within the user's scope

Examples

Example 1: Count - no criteria (GET)

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/optionprofile
/"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>30</count>
</ServiceResponse>
```

Example 2: Count - criteria (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/optionprofile
/" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
    <filters>
        <Criteria field="id"
operator="IN">832265669,832295669,832285669</Criteria>
        <Criteria field="name" operator="CONTAINS">OP</Criteria>
        <Criteria field="tags" operator="NONE"></Criteria>
        <Criteria field="createdDate" operator="LESSER">2014-09-
09</Criteria>
        <Criteria field="updatedDate" operator="LESSER">2014-09-
09</Criteria>
    </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
</ServiceResponse>
```

Search option profiles

Returns a list of option profiles which are in the user's scope. Action logs are not included in the output.

URL: https://qualysapi.qualys.com/qps/rest/3.0/search/was/optionprofile

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND..

id (Integer)	Filter by option profile ID
name (Text)	Filter by option profile name
tags	Filter by tags applied
tags.id (Integer)	Filter by tag ID applied
tags.name (Text)	Filter by tag name applied
createdDate (Date)	Filter by created date (UTC date/time format)
updatedDate (Date)	Filter by last updated date (UTC date/time format)
usedByWebApps (Boolean with operator: EQUALS or NOT EQUALS)	Filter profiles if used/not used by web applications
usedBySchedules (Boolean with operator: EQUALS or NOT EQUALS)	Filter profiles if used/not used by scan schedules
owner.id (Long with operator: EQUALS, IN, NOT EQUALS, GREATER or LESSER)	Filter profiles based on owner's user ID
owner.name (text with operator: CONTAINS, EQUALS or NOT EQUALS)	Filter profiles based on owner's full name (first and last)
owner.username (text with operator: CONTAINS, EQUALS or NOT EQUALS)	Filter profiles based on owner's username (like acme_ab3)

Allowed Operators	
Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN
Boolean	(true/false) EQUALS, NOT EQUALS

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Output includes option profiles within the user's scope

Example

Search - criteria (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/optionprofil
e/" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
REQUEST:
<ServiceRequest>
    <filters>
        <Criteria field="id"
operator="IN">832265669,832295669,832285669</Criteria>
        <Criteria field="name" operator="CONTAINS">OP</Criteria>
        <Criteria field="tags" operator="NONE"></Criteria>
        <Criteria field="createdDate" operator="LESSER">2014-09-
09</Criteria>
        <Criteria field="updatedDate" operator="LESSER">2014-09-
09</Criteria>
    </filters>
```

```
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <OptionProfile>
      <id>832285669</id>
      <name><![CDATA[My Option Profile]]></name>
      <owner>
        <id>8792415669</id>
        <username>acme_ww</username>
        <firstName><![CDATA[Walter]]></firstName>
        <lastName><![CDATA[White]]></lastName>
      </owner>
      <tags>
        <count>0</count>
      </tags>
      <createdDate>2014-09-08T23:16:07Z</createdDate>
      <updatedDate>2014-09-08T23:16:07Z</updatedDate>
    </OptionProfile>
  </data>
</ServiceResponse>
```

Get details for an option profile

View details for an option profile which is in the user's scope. See "Search option profiles" to find a record ID to use as input.

URL: https://qualysapi.qualys.com/qps/rest/3.0/get/was/optionprofile/<id>

Methods allowed: GET

Input

The element "id" (Integer) is required, where "id" identifies an option profile.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
Option profile must be within the user's scope

Example

View details (GET)

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/optionprofile/8
32265669"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <OptionProfile>
      <id>832265669</id>
      <name><![CDATA[My Option Profile]]></name>
      <owner>
        <id>8792415669</id>
        <username>acme_ww</username>
```

```
  <firstName><![CDATA[Walter]]></firstName>
  <lastName><![CDATA[White]]></lastName>
</owner>
<isDefault>false</isDefault>
<tags>
  <count>0</count>
</tags>
<formSubmission>BOTH</formSubmission>
<maxCrawlRequests>300</maxCrawlRequests>
<parameterSet>
  <id>0</id>
  <name><![CDATA[Initial Parameters]]></name>
</parameterSet>
<ignoreBinaryFiles>false</ignoreBinaryFiles>
<performance>LOW</performance>
<bruteforceOption>MINIMAL</bruteforceOption>
<comments>
  <count>2</count>
  <list>
    <Comment>
      <contents><![CDATA[some comments]]></contents>
      <author>
        <id>200639085669</id>
        <username>acme_ww</username>
      </author>
    </Comment>
    <Comment>
      <contents><![CDATA[some more comments]]></contents>
      <author>
        <id>200639085669</id>
        <username>acme_ww</username>
      </author>
    </Comment>
  </list>
</comments>
<sensitiveContent>
  <creditCardNumber>false</creditCardNumber>
  <socialSecurityNumber>false</socialSecurityNumber>
</sensitiveContent>
<createdDate>2014-09-08T22:03:01Z</createdDate>
<createdBy>
```

```
          <id>8792415669</id>
          <username>acme_ww</username>
          <firstName><![CDATA[Walter]]></firstName>
          <lastName><![CDATA[White]]></lastName>
        </createdBy>
        <updatedDate>2014-09-08T23:18:28Z</updatedDate>
        <updatedBy>
          <id>8792415669</id>
          <username>acme_ww</username>
          <firstName><![CDATA[Walter]]></firstName>
          <lastName><![CDATA[White]]></lastName>
        </updatedBy>
      </OptionProfile>
    </data>
  </ServiceResponse>
```

Create a new option profile

Create a new option profile.

URL: https://qualysapi.qualys.com/qps/rest/3.0/create/was/optionprofile

Methods allowed: POST

Input

Required elements are “name” and “OptionProfile”. Additional elements are optional - see examples below.

Permissions

User must have the WAS application enabled
User must have “API Access” permission
User must have “Create Option Profile” permission

Example

Example 1: Create - minimum criteria (POST)

Create a new option profile with the name “My Option Profile - with defaults”. The default option profile settings are assigned automatically.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/optionprofil
e/" < file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
   <data>
      <OptionProfile>
        <name><![CDATA[My Option Profile - with defaults]]></name>
      </OptionProfile>
   </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <OptionProfile>
      <id>832265669</id>
      <name><![CDATA[My Option Profile - with defaults]]></name>
      <owner>
        <id>8792415669</id>
        <username>acme_as</username>
        <firstName><![CDATA[Alex]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
      <isDefault>false</isDefault>
      <tags>
        <count>0</count>
      </tags>
      <formSubmission>BOTH</formSubmission>
      <maxCrawlRequests>300</maxCrawlRequests>
      <parameterSet>
        <id>0</id>
        <name><![CDATA[Initial Parameters]]></name>
      </parameterSet>
      <ignoreBinaryFiles>false</ignoreBinaryFiles>
      <performance>LOW</performance>
      <bruteforceOption>MINIMAL</bruteforceOption>
      <comments>
        <count>0</count>
      </comments>
      <sensitiveContent>
        <creditCardNumber>false</creditCardNumber>
        <socialSecurityNumber>false</socialSecurityNumber>
      </sensitiveContent>
      <createdDate>2014-09-08T22:03:01Z</createdDate>
      <createdBy>
        <id>8792415669</id>
```

```
        <username>acme_as</username>
        <firstName><![CDATA[Alex]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </createdBy>
      <updatedDate>2014-09-08T22:03:01Z</updatedDate>
      <updatedBy>
        <id>8792415669</id>
        <username>acme_as</username>
        <firstName><![CDATA[Alex]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
    </OptionProfile>
  </data>
</ServiceResponse>
```

Example 2: Create - all criteria (POST)

Create a new option profile with the name “My Option Profile - All Fields”. The "name" setting is required in the request data, other settings are optional.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/optionprofil
e/" < file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
   <data>
      <OptionProfile>
         <name><![CDATA[My Option Profile - All Fields]]></name>
         <formSubmission>BOTH</formSubmission>
         <maxCrawlRequests>200</maxCrawlRequests>
         <performance>LOW</performance>
         <bruteforceOption>USER_DEFINED</bruteforceOption>
         <parameterSet><id>15669</id></parameterSet>
         <isDefault>true</isDefault>
         <ignoreBinaryFiles>true</ignoreBinaryFiles>
         <userAgent><![CDATA[Mozilla/5.0 (Windows NT 6.2; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.116
```

```
Safari/537.36]]></userAgent>
         <tags><set><Tag><id>75521225669</id></Tag></set></tags>
         <sensitiveContent>
         <customContents>zip code</customContents>
         </sensitiveContent>
         <comments>
            <set>
               <Comment>
<contents><![CDATA[Some Comment]]></contents>
               </Comment>
            </set>
         </comments>
         <bruteforceList>
            <id>74005669</id>
         </bruteforceList>
         <detection>
            <includedSearchLists>
               <set>
                   <SearchList>
                        <id>3496185669</id>
                   </SearchList>
               </set>
            </includedSearchLists>
            <excludedSearchLists>
               <set>
                  <SearchList>
                     <id>3496175669</id>
                  </SearchList>
                  <SearchList>
                     <id>3496165669</id>
                  </SearchList>
               </set>
            </excludedSearchLists>
         </detection>
      </OptionProfile>
   </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <OptionProfile>
      <id>832275669</id>
      <name><![CDATA[My Option Profile - All Fields]]></name>
      <owner>
        <id>8792415669</id>
        <username>acme_cg</username>
        <firstName><![CDATA[Cindy]]></firstName>
        <lastName><![CDATA[Green]]></lastName>
      </owner>
      <isDefault>true</isDefault>
      <tags>
        <count>1</count>
        <list>
          <Tag>
            <id>75521225669</id>
            <name><![CDATA[Business Units]]></name>
          </Tag>
        </list>
      </tags>
      <formSubmission>BOTH</formSubmission>
      <maxCrawlRequests>200</maxCrawlRequests>
      <userAgent><![CDATA[Mozilla/5.0 (Windows NT 6.2; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.116
Safari/537.36]]></userAgent>
      <parameterSet>
        <id>15669</id>
        <name><![CDATA[Custom Parameters]]></name>
      </parameterSet>
      <ignoreBinaryFiles>true</ignoreBinaryFiles>
      <performance>LOW</performance>
      <bruteforceOption>USER_DEFINED</bruteforceOption>
      <bruteforceList>
```

```
    <id>74005669</id>
    <name><![CDATA[BFL]]></name>
  </bruteforceList>
  <detection>
    <includedSearchLists>
      <count>1</count>
      <list>
        <SearchList>
          <id>3496185669</id>
        </SearchList>
      </list>
    </includedSearchLists>
    <excludedSearchLists>
      <count>2</count>
      <list>
        <SearchList>
          <id>3496175669</id>
        </SearchList>
        <SearchList>
          <id>3496165669</id>
        </SearchList>
      </list>
    </excludedSearchLists>
  </detection>
  <comments>
    <count>1</count>
    <list>
      <Comment>
        <contents><![CDATA[Some Comment]]></contents>
      </Comment>
    </list>
  </comments>
  <sensitiveContent>
    <creditCardNumber>false</creditCardNumber>
    <socialSecurityNumber>false</socialSecurityNumber>
    <customContents>zip code</customContents>
  </sensitiveContent>
  <createdDate>2014-09-08T22:31:06Z</createdDate>
  <createdBy>
    <id>8792415669</id>
    <username>acme_cg</username>
```

```
        <firstName><![CDATA[Cindy]]></firstName>
        <lastName><![CDATA[Green]]></lastName>
      </createdBy>
      <updatedDate>2014-09-08T22:31:07Z</updatedDate>
      <updatedBy>
        <id>8792415669</id>
        <username>acme_cg</username>
        <firstName><![CDATA[Cindy]]></firstName>
        <lastName><![CDATA[Green]]></lastName>
      </updatedBy>
    </OptionProfile>
  </data>
</ServiceResponse>
```

Update an option profile

Update an option profile which is in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofile/<id>

Methods allowed: POST

Input

The element "id" (Integer) is required, where "id" identifies an option profile. Additional elements are optional and must be supplied in POST XML data. At least one of the following elements must be set: name, isDefault, owner, tags, formSubmission, maxCrawlRequests, userAgent, parameterSet, ignoreBinaryFiles, performance, bruteforceOption, bruteforceList, numberOfAttempts, detection, sensitiveContent, comments.

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have the "Update Option Profile" permission

Example

Example 1: Update - minimum criteria (POST)

Change the option profile name to "Update Option Profile - title" for option profile ID 832265669.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofil
e/832265669" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
   <data>
      <OptionProfile>
         <name><![CDATA[Update Option Profile - title ]]></name>
```

```
      </OptionProfile>
   </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <OptionProfile>
      <id>832265669</id>
    </OptionProfile>
  </data>
</ServiceResponse>
```

Example 2: Update - multiple settings (POST)

Update multiple option profile settings for option profile ID 832275669.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofil
e/832275669" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
   <data>
      <OptionProfile>
         <name><![CDATA[My Option Profile - All Fields]]></name>
         <formSubmission>BOTH</formSubmission>
         <maxCrawlRequests>100</maxCrawlRequests>
         <performance>HIGH</performance>
         <bruteforceOption>USER_DEFINED</bruteforceOption>
         <parameterSet><id>15669</id></parameterSet>
```

```
         <isDefault>false</isDefault>
         <ignoreBinaryFiles>false</ignoreBinaryFiles>
         <userAgent><![CDATA[Mozilla/5.0 (Windows NT 6.2; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.116
Safari/537.36]]></userAgent>
         <tags><set><Tag><id>75521225669</id></Tag></set></tags>
         <sensitiveContent>
            <customContents>zip code</customContents>
         </sensitiveContent>
         <comments>
            <set>
               <Comment>
                  <contents><![CDATA[Comment 2]]></contents>
               </Comment>
            </set>
         </comments>
         <bruteforceList>
            <id>74005669</id>
         </bruteforceList>
         <detection>
            <includedSearchLists>
               <set>
                  <SearchList>
                     <id>3496185669</id>
                  </SearchList>
               </set>
            </includedSearchLists>
            <excludedSearchLists>
               <set>
                  <SearchList>
                     <id>3496175669</id>
                  </SearchList>
                  <SearchList>
                     <id>3496165669</id>
                  </SearchList>
               </set>
            </excludedSearchLists>
         </detection>
      </OptionProfile>
   </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <OptionProfile>
      <id>832275669</id>
    </OptionProfile>
  </data>
  </ServiceRequest>
```

Delete an option profile

Delete an option profile that is in the user's scope. Upon success, the output is a list of IDs for the option profiles that were deleted.

URL: https://qualysapi.qualys.com/qps/rest/3.0/delete/was/optionprofile/<id>
or
https://qualysapi.qualys.com/qps/rest/3.0/delete/was/optionprofile

Methods allowed: POST

Input

Optional elements are used to retrieve option profiles to delete. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

name (Text)	updatedDate (Date)
owner (Text)	usedByWebApps (Boolean)
tags	usedBySchedules (Boolean)
createdDate (Date)	

Permissions

User must have the WAS application enabled
User must have "API Access" permission
User must have the "Delete Option Profile" permission

Examples

Example 1: Delete specific option profile (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/optionprofil
e/834275669"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <OptionProfile>
      <id>834275669</id>
    </OptionProfile>
  </data>
</ServiceResponse>
```

Example 2: Delete multiple option profiles (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/optionprofil
e/" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
    <filters>
        <Criteria field="name" operator="CONTAINS">OP</Criteria>
        <Criteria field="updatedDate" operator="LESSER">2014-09-
09</Criteria>
    </filters>
</ServiceRequest>
```

Response:

```
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>10</count>
```

```
<data>
  <OptionProfile>
    <id>712265669</id>
  </OptionProfile>
  <OptionProfile>
    <id>752265669</id>
  </OptionProfile>
  <OptionProfile>
    <id>752275669</id>
  </OptionProfile>
  <OptionProfile>
    <id>754265669</id>
  </OptionProfile>
  <OptionProfile>
    <id>812685669</id>
  </OptionProfile>
  <OptionProfile>
    <id>824295669</id>
  </OptionProfile>
  <OptionProfile>
    <id>824305669</id>
  </OptionProfile>
  <OptionProfile>
    <id>830265669</id>
  </OptionProfile>
  <OptionProfile>
    <id>830275669</id>
  </OptionProfile>
  <OptionProfile>
    <id>830285669</id>
  </OptionProfile>
    </data>
</ServiceResponse>
```

CHAPTER 9

Finding API

The WAS Finding API provides a suite of API functions lets you manage the findings (detections) returned from your web application scans.

These operations are available:

Current finding count

Search findings

Get details of a finding

Ignore Findings

Activate Findings

Current finding count

Returns the total number of findings on web application(s) in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/count/was/finding

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

id (Integer)	ignoredDate (Date)
qid (Integer)	ignoredReason (Keyword: FALSE_POSITIVE, RISK_ACCEPTED or NOT_APPLICABLE)
name (Text)	group (Keyword: XSS, SQL, INFO, PATH, CC, SSN_US or CUSTOM)
type (Keyword: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED)	owasp.name (Text)
url (Text)	owasp.code (Integer)
webapp.tags.id (Integer)	wasc.name (Text)
webapp.tags.name (Text)	wasc.code (Integer)
status (Keyword) NEW, ACTIVE or REOPENED	cwe.id (Integer)
patch (Integer-Long)	firstDetectedDate (Date)
Webapp.id (Integer)	lastDetectedDate (Date)
webapps.name (Text)	lastTestedDate (Date)
severity (Integer)	timesDetected (Integer)
externalRef (String) Tip - Use operator IS EMPTY for findings with empty external references	

Allowed Operators	
Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS, IS EMPTY (use with externalRef only)
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN
Boolean	(true/false) EQUALS, NOT EQUALS

Permissions

The WAS application must be enabled in the user's account
User must have "API Access" permission
Count includes findings on web applications within the user's scope

Example

Example 1: Count - no criteria (GET)

Return the number (count) of all findings in the user's scope.

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/finding/"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2815</count>
</ServiceResponse>
```

Example 2: Count - criteria (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/finding/" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
       <filters>
         <Criteria field="type"
operator="EQUALS">VULNERABILITY</Criteria>
         <Criteria field="severity" operator="EQUALS">5</Criteria>
         <Criteria field="status" operator="IN">NEW, ACTIVE,
REOPENED</Criteria>
       </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>41</count>
</ServiceResponse>
```

Search findings

Returns list of findings (vulnerabilities, sensitive contents, information gathered) found in web applications which are in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/search/was/finding

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

id (Integer)	ignoredDate (Date)
qid (Integer)	ignoredReason (Keyword: FALSE_POSITIVE, RISK_ACCEPTED or NOT_APPLICABLE)
name (Text)	group (Keyword: XSS, SQL, INFO, PATH, CC, SSN_US or CUSTOM)
type (Keyword: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED)	owasp.name (Text)
url (Text)	owasp.code (Integer)
webapp.tags.id (Integer)	wasc.name (Text)
webapp.tags.name (Text)	wasc.code (Integer)
status (Keyword: NEW, ACTIVE or REOPENED)	cwe.id (Integer)
patch (Integer-Long)	firstDetectedDate (Date)
Webapp.id (Integer)	lastDetectedDate (Date)
webapps.name (Text)	lastTestedDate (Date)
severity (Integer)	timesDetected (Integer)
externalRef (String) Tip - Use operator IS EMPTY for findings with empty external references	

Allowed Operators	
Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS, IS EMPTY (use with externalRef only)
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN

Permissions

The WAS application must be enabled in the user's account
User must have "API Access" permission
Output contains findings on web applications within the user's scope

Examples

Example 1: Search - no criteria (POST)

Return a list of all findings in the user's scope.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X POST
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/finding/"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>100</count>
  <hasMoreRecords>true</hasMoreRecords>
  <lastId>39271</lastId>
  <data>
    <Finding>
      <id>5665</id>
      <qid>150016</qid>
      <name><![CDATA[Sensitive Content In HTML]]></name>
```

```
      <type>SENSITIVE_CONTENT</type>
      <severity>2</severity>
<url><![CDATA[http://funkytown.vuln.qa.qualys.com/cassium/login/1/
login.php]]></url>
      <status>NEW</status>
      <lastDetectedDate>2014-02-05T23:38:19Z</lastDetectedDate>
      <lastTestedDate>2014-02-05T23:38:19Z</lastTestedDate>
      <webApp>
        <id>95729</id>
        <name><![CDATA[WA3 - SC]]></name>
        <url><![CDATA[http://funkytown.vuln.qa.qualys.com]]></url>
      </webApp>
    </Finding>
    <Finding>
      <id>5666</id>
      <qid>150016</qid>
      <name><![CDATA[Sensitive Content In HTML]]></name>
      <type>SENSITIVE_CONTENT</type>
      <severity>2</severity>
<url><![CDATA[http://funkytown.vuln.qa.qualys.com/cassium/login/2/
login2.php]]></url>
      <status>NEW</status>
      <lastDetectedDate>2014-02-05T23:38:19Z</lastDetectedDate>
      <lastTestedDate>2014-02-05T23:38:19Z</lastTestedDate>
      <webApp>
        <id>95729</id>
        <name><![CDATA[WA3 - SC]]></name>
        <url><![CDATA[http://funkytown.vuln.qa.qualys.com]]></url>
      </webApp>
    </Finding>
...
```

Example 2: Search - criteria (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/finding/" <
file.xml
```

Note: “file.xml” contains the request POST data.

Request POST Data:

```
<ServiceRequest>
   <preferences>
      <limitResults>1000</limitResults>
      <verbose>true</verbose>
   </preferences>
   <filters>
      <Criteria field="id" operator="EQUALS">1137289</Criteria>
   </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <Finding>
      <id>1137289</id>
      <qid>150013</qid>
      <name><![CDATA[Browser-Specific Cross-Site Scripting
Vulnerabilities]]></name>
      <type>VULNERABILITY</type>
      <group>XSS</group>
      <cwe>
        <count>1</count>
        <list>
          <long>79</long>
        </list>
      </cwe>
      <owasp>
        <count>1</count>
        <list>
          <OWASP>
            <name><![CDATA[Cross-Site Scripting (XSS)]]></name>
<url><![CDATA[https://www.owasp.org/index.php/Top_10_2013-A3-
```

```
Cross-Site_Scripting_(XSS)]]></url>
            <code>3</code>
          </OWASP>
        </list>
      </owasp>
      <wasc>
        <count>1</count>
        <list>
          <WASC>
            <name><![CDATA[Cross-Site Scripting]]></name>
<url><![CDATA[http://projects.webappsec.org/w/page/13246920/WASC]]
></url>
           <code>8</code>
          </WASC>
        </list>
      </wasc>
      <param><![CDATA[accountcorp]]></param>
      <resultList>
        <count>1</count>
        <list>
          <Result>
            <authentication>false</authentication>
            <accessPath>
              <count>1</count>
              <list>
                <Url><![CDATA[https://10.10.26.238/]]></Url>
              </list>
            </accessPath>
            <payloads>
              <count>5</count>
              <list>
                <PayloadInstance>
                  <payload><![CDATA[@PATH@@FILE@.test2 <![CDATA[
TEST ] ]>]]></payload>
                    <request>
                      <method><![CDATA[GET]]></method>
<link><![CDATA[http://10.10.26.238/?accountcorp=%3Cscript%20src%3D
http%3A%2F%2Flocalhost%2Fj%20]]></link>
<headers><![CDATA[UmVmZXJlcjogaHR0cHM6Ly8xMC4xMC4yNi4yMzgvDQpDb29r
aWU6IGNvb2tpZTM9Y29va2lldGhyZWU7IGNvb2tpZTI9Y29va2lldHdvOyBjb29raW
UxPWNvb2tpZW9uZTsgUEhQU0VTU0lEPWQ3YTBlYWI5NzNhOTIyNWE2NjkzODljMjlh
ZWVhYTRlOw0K]]></headers>
```

```
                  </request>
                 <response><![CDATA[HTTP/1.1 <![CDATA[ TEST2 ] ]>
200 OK]]></response>
                  <payloadResponce>
                    <offset>232</offset>
                    <length>36</length>
                  </payloadResponce>
                </PayloadInstance>
                <PayloadInstance>
                  <payload><![CDATA[@PATH@@FILE@.test2 <![CDATA[
TEST ] ]>]]></payload>
                  <request>
                    <method><![CDATA[GET]]></method>
<link><![CDATA[http://10.10.26.238/?accountcorp=%3Cscript%20src%3D
http%3A%2F%2Flocalhost%2Fj%20]]></link>
<headers><![CDATA[UmVmZXJlcjogaHR0cHM6Ly8xMC4xMC4yNi4yMzgvDQpDb29r
aWU6IGNvb2tpZTM9Y29va2lldGhyZWU7IGNvb2tpZTI9Y29va2lldHdvOyBjb29raW
UxPWNvb2tpZW9uZTsgUEhQU0VTU0lEPTBjZWY2OTUzNmI5M2MyYTQ3ZmMyYzI3NWJm
NjI4MDc3Ow0K]]></headers>
                  </request>
                 <response><![CDATA[HTTP/1.1 <![CDATA[ TEST2 ] ]>
200 OK]]></response>
                  <payloadResponce>
                    <offset>232</offset>
                    <length>36</length>
                  </payloadResponce>
                </PayloadInstance>
                <PayloadInstance>
                  <payload><![CDATA[@PATH@@FILE@.test2 <![CDATA[
TEST ] ]>]]></payload>
                  <request>
                    <method><![CDATA[GET]]></method>
<link><![CDATA[http://10.10.26.238/?accountcorp=%3Cscript%20src%3D
http%3A%2F%2Flocalhost%2Fj%20]]></link>
<headers><![CDATA[UmVmZXJlcjogaHR0cHM6Ly8xMC4xMC4yNi4yMzgvDQpDb29r
aWU6IGNvb2tpZTM9Y29va2lldGhyZWU7IGNvb2tpZTI9Y29va2lldHdvOyBjb29raW
UxPWNvb2tpZW9uZTsgUEhQU0VTU0lEPWFmMGJkMmIxOGE0MjgyNDRhYWYxNzZiMmI1
ODcwMTY1Ow0K]]></headers>
                  </request>
                 <response><![CDATA[HTTP/1.1 <![CDATA[ TEST2 ] ]>
200 OK]]></response>
                  <payloadResponce>
                    <offset>232</offset>
```

```
                    <length>36</length>
                  </payloadResponce>
                </PayloadInstance>
                <PayloadInstance>
                  <payload><![CDATA[@PATH@@FILE@.test2 <![CDATA[
TEST ] ]>]]></payload>
                  <request>
                    <method><![CDATA[GET]]></method>
<link><![CDATA[http://10.10.26.238/?accountcorp=%3Cscript%20src%3D
http%3A%2F%2Flocalhost%2Fj%20]]></link>
<headers><![CDATA[UmVmZXJlcjogaHR0cHM6Ly8xMC4xMC4yNi4yMzgvDQpDb29r
aWU6IGNvb2tpZTM9Y29va2lldGhyZWU7IGNvb2tpZTI9Y29va2lldHdvOyBjb29raW
UxPWNvb2tpZW9uZTsgUEhQU0VTU0lEPTEyY2I1YzhhNTYwMDRjYWM2MDY2N2M0ZTg0
Y2VmZWNjOw0K]]></headers>
                  </request>
                  <response><![CDATA[HTTP/1.1 <![CDATA[ TEST2 ] ]>
200 OK]]></response>
                  <payloadResponce>
                    <offset>232</offset>
                    <length>36</length>
                  </payloadResponce>
                </PayloadInstance>
                <PayloadInstance>
                  <payload><![CDATA[@PATH@@FILE@.test2 <![CDATA[
TEST ] ]>]]></payload>
                  <request>
                    <method><![CDATA[GET]]></method>
<link><![CDATA[http://10.10.26.238/?accountcorp=%3Cscript%20src%3D
http%3A%2F%2Flocalhost%2Fj%20]]></link>
<headers><![CDATA[UmVmZXJlcjogaHR0cHM6Ly8xMC4xMC4yNi4yMzgvDQpDb29r
aWU6IGNvb2tpZTM9Y29va2lldGhyZWU7IGNvb2tpZTI9Y29va2lldHdvOyBjb29raW
UxPWNvb2tpZW9uZTsgUEhQU0VTU0lEPTJjYTMxOTk5Y2VlYTBmMzIzNDRhMzgyNzEz
MWJmYTAxOw0K]]></headers>
                  </request>
                  <response><![CDATA[HTTP/1.1 <![CDATA[ TEST2 ] ]>
200 OK]]></response>
                  <payloadResponce>
                    <offset>232</offset>
                    <length>36</length>
                  </payloadResponce>
                </PayloadInstance>
              </list>
            </payloads>
```

```
          </Result>
        </list>
      </resultList>
      <severity>5</severity>
<url><![CDATA[http://10.10.26.238/?accountcorp=%3Cscript%20src%3Dh
ttp%3A%2F%2Flocalhost%2Fj%20]]></url>
      <status>ACTIVE</status>
      <firstDetectedDate>2014-12-01T19:43:09Z</firstDetectedDate>
      <lastDetectedDate>2014-12-05T19:42:38Z</lastDetectedDate>
      <lastTestedDate>2014-12-05T19:42:38Z</lastTestedDate>
      <timesDetected>5</timesDetected>
      <webApp>
        <id>1930868</id>
        <name><![CDATA[10.10.26.238:443 (subuser form
sitemap)]]></name>
        <url><![CDATA[https://10.10.26.238:443]]></url>
      </webApp>
      <isIgnored>false</isIgnored>
      <externalRef><![CDATA[aaaaabbbbbccccc]]></externalRef>
    </Finding>
  </data>
</ServiceResponse>
```

Get details of a finding

View details for a finding on a web application which is in the user's scope. See "Search findings" to find a record ID to use as input.

URL: https://qualysapi.qualys.com/qps/rest/3.0/get/was/finding/id

Methods allowed: GET

Input

The element "id" (Integer) is required, where "id" identifies a finding (WebAppVuln, WebAppIg, or WebAppSensitiveContent).

Permissions

The WAS application must be enabled in the user's account
User must have "API Access" permission
Web application must be within the user's scope

Example

View details (GET)

Request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/finding/1137289
"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Finding>
      <id>1137289</id>
      <qid>150013</qid>
      <name><![CDATA[Browser-Specific Cross-Site Scripting
Vulnerabilities]]></name>
```

```
        <type>VULNERABILITY</type>
        <group>XSS</group>
        <cwe>
          <count>1</count>
          <list>
            <long>79</long>
          </list>
        </cwe>
        <owasp>
          <count>1</count>
          <list>
            <OWASP>
              <name><![CDATA[Cross-Site Scripting (XSS)]]></name>

<url><![CDATA[https://www.owasp.org/index.php/Top_10_2013-A3-
Cross-Site_Scripting_(XSS)]]></url>
              <code>3</code>
            </OWASP>
          </list>
        </owasp>
        <wasc>
          <count>1</count>
          <list>
            <WASC>
             <name><![CDATA[Cross-Site Scripting]]></name>
<url><![CDATA[http://projects.webappsec.org/w/page/13246920/WASC]]
></url>
              <code>8</code>
            </WASC>
          </list>
        </wasc>
        <param><![CDATA[accountcorp]]></param>
        <resultList>
          <count>1</count>
          <list>
            <Result>
              <authentication>false</authentication>
              <accessPath>
                <count>1</count>
                <list>
                  <Url><![CDATA[https://10.10.26.238/]]></Url>
                </list>
```

```
            </accessPath>
            <payloads>
              <count>5</count>
              <list>
                <PayloadInstance>
                  <payload><![CDATA[@PATH@@FILE@.test2 <![CDATA[
TEST ] ]>]]></payload>
                   <request>
                  <method><![CDATA[GET]]></method>
<link><![CDATA[http://10.10.26.238/?accountcorp=%3Cscript%20src%3D
http%3A%2F%2Flocalhost%2Fj%20]]></link>
<headers><![CDATA[UmVmZXJlcjogaHR0cHM6Ly8xMC4xMC4yNi4yMzgvDQpDb29r
aWU6IGNvb2tpZTM9Y29va2lldGhyZWU7IGNvb2tpZTI9Y29va2lldHdvOyBjb29raW
UxPWNvb2tpZW9uZTsgUEhQU0VTU0lEPWQ3YTBlYWI5NzNhOTIyNWE2NjkzODljMjlh
ZWVhYTRlOw0K]]></headers>
                   </request>
                  <response><![CDATA[HTTP/1.1 <![CDATA[ TEST2 ] ]>
200 OK]]></response>
                   <payloadResponce>
                     <offset>232</offset>
                     <length>36</length>
                   </payloadResponce>
                 </PayloadInstance>
                 <PayloadInstance>
                   <payload><![CDATA[@PATH@@FILE@.test2 <![CDATA[
TEST ] ]>]]></payload>
                   <request>
                  <method><![CDATA[GET]]></method>
<link><![CDATA[http://10.10.26.238/?accountcorp=%3Cscript%20src%3D
http%3A%2F%2Flocalhost%2Fj%20]]></link>
<headers><![CDATA[UmVmZXJlcjogaHR0cHM6Ly8xMC4xMC4yNi4yMzgvDQpDb29r
aWU6IGNvb2tpZTM9Y29va2lldGhyZWU7IGNvb2tpZTI9Y29va2lldHdvOyBjb29raW
UxPWNvb2tpZW9uZTsgUEhQU0VTU0lEPTBjZWY2OTUzNmI5M2MyYTQ3ZmMyYzI3NWJm
NjI4MDc3Ow0K]]></headers>
                   </request>
                  <response><![CDATA[HTTP/1.1 <![CDATA[ TEST2 ] ]>
200 OK]]></response>
                   <payloadResponce>
                     <offset>232</offset>
                     <length>36</length>
                   </payloadResponce>
                 </PayloadInstance>
                 <PayloadInstance>
```

```
                  <payload><![CDATA[@PATH@@FILE@.test2 <![CDATA[
TEST ] ]>]]></payload>
                  <request>
                 <method><![CDATA[GET]]></method>
<link><![CDATA[http://10.10.26.238/?accountcorp=%3Cscript%20src%3D
http%3A%2F%2Flocalhost%2Fj%20]]></link>
<headers><![CDATA[UmVmZXJlcjogaHR0cHM6Ly8xMC4xMC4yNi4yMzgvDQpDb29r
aWU6IGNvb2tpZTM9Y29va2lldGhyZWU7IGNvb2tpZTI9Y29va2lldHdvOyBjb29raW
UxPWNvb2tpZW9uZTsgUEhQU0VTU0lEPWFmMGJkMmIxOGE0MjgyNDRhYWYxNzZiMmI1
ODcwMTY1Ow0K]]></headers>
                  </request>
                 <response><![CDATA[HTTP/1.1 <![CDATA[ TEST2 ] ]>
200 OK]]></response>
                  <payloadResponce>
                    <offset>232</offset>
                    <length>36</length>
                  </payloadResponce>
                </PayloadInstance>
                <PayloadInstance>
                  <payload><![CDATA[@PATH@@FILE@.test2 <![CDATA[
TEST ] ]>]]></payload>
                  <request>
                 <method><![CDATA[GET]]></method>
<link><![CDATA[http://10.10.26.238/?accountcorp=%3Cscript%20src%3D
http%3A%2F%2Flocalhost%2Fj%20]]></link>
<headers><![CDATA[UmVmZXJlcjogaHR0cHM6Ly8xMC4xMC4yNi4yMzgvDQpDb29r
aWU6IGNvb2tpZTM9Y29va2lldGhyZWU7IGNvb2tpZTI9Y29va2lldHdvOyBjb29raW
UxPWNvb2tpZW9uZTsgUEhQU0VTU0lEPTEyY2I1YzhhNTYwMDRjYWM2MDY2N2M0ZTg0
Y2VmZWNjOw0K]]></headers>
                  </request>
                 <response><![CDATA[HTTP/1.1 <![CDATA[ TEST2 ] ]>
200 OK]]></response>
                  <payloadResponce>
                    <offset>232</offset>
                    <length>36</length>
                  </payloadResponce>
                </PayloadInstance>
                <PayloadInstance>
                  <payload><![CDATA[@PATH@@FILE@.test2 <![CDATA[
TEST ] ]>]]></payload>
                  <request>
                 <method><![CDATA[GET]]></method>
<link><![CDATA[http://10.10.26.238/?accountcorp=%3Cscript%20src%3D
```

```
http%3A%2F%2Flocalhost%2Fj%20]]></link>
<headers><![CDATA[UmVmZXJlcjogaHR0cHM6Ly8xMC4xMC4yNi4yMzgvDQpDb29r
aWU6IGNvb2tpZTM9Y29va2lldGhyZWU7IGNvb2tpZTI9Y29va2lldHdvOyBjb29raW
UxPWNvb2tpZW9uZTsgUEhQU0VTU0lEPTJjYTMxOTk5Y2VlYTBmMzIzNDRhMzgyNzEz
MWJmYTAxOw0K]]></headers>
                  </request>
                  <response><![CDATA[HTTP/1.1 <![CDATA[ TEST2 ] ]>
200 OK]]></response>
                  <payloadResponce>
                    <offset>232</offset>
                    <length>36</length>
                  </payloadResponce>
                </PayloadInstance>
              </list>
            </payloads>
          </Result>
        </list>
      </resultList>
      <severity>5</severity>
<url><![CDATA[http://10.10.26.238/?accountcorp=%3Cscript%20src%3Dh
ttp%3A%2F%2Flocalhost%2Fj%20]]></url>
      <status>ACTIVE</status>
      <firstDetectedDate>2014-12-01T19:43:09Z</firstDetectedDate>
      <lastDetectedDate>2014-12-05T19:42:38Z</lastDetectedDate>
      <lastTestedDate>2014-12-05T19:42:38Z</lastTestedDate>
      <timesDetected>5</timesDetected>
      <webApp>
        <id>1930868</id>
        <name><![CDATA[10.10.26.238:443 (subuser form
sitemap)]]></name>
        <url><![CDATA[https://10.10.26.238:443]]></url>
      </webApp>
      <patch>18007</patch>
      <isIgnored>false</isIgnored>
      <externalRef><![CDATA[aaaaabbbbbccccc]]></externalRef>
    </Finding>
  </data>
</ServiceResponse>
```

Ignore Findings

Ignore findings for a web application which is in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/ignore/was/finding

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

id (Integer)	ignoredReason (Keyword: FALSE_POSITIVE, RISK_ACCEPTED or NOT_APPLICABLE)
qid (Integer)	group (Keyword: XSS, SQL, INFO, PATH, CC, SSN_US or CUSTOM)
name (Text)	owasp.name (Text)
type (Keyword: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED)	owasp.code (Integer)
url (Text)	wasc.name (Text)
webapp.tags.id (Integer)	wasc.code (Integer)
webapp.tags.name (Text)	cwe.id (Integer)
status (Keyword: NEW, ACTIVE or REOPENED)	firstDetectedDate (Date)
Webapp.id (Integer)	lastDetectedDate (Date)
webapps.name (Text)	lastTestedDate (Date)
severity (Integer)	timesDetected (Integer)
ignoredDate (Date)	

Permissions

The WAS application must be enabled in the user's account
User must have "API Access" permission
Web application must be within the user's scope
User must have "Ignore Vulnerabilities" permission

Examples

Ignore specific finding (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/ignore/was/finding/1645
195669"
```

Request POST Data:

```
<ServiceRequest>
        <data>
           <Finding>
                <id>1645195669</id>
               <ignoredReason>FALSE_POSITIVE</ignoredReason>
                <ignoredComment>test</ignoredComment>
           </Finding>
        </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Finding>
      <id>1645195669</id>
    </Finding>
  </data>
</ServiceResponse>
```

Activate Findings

Activate ignored findings for a web application which is in the user's scope.

URL: https://qualysapi.qualys.com/qps/rest/3.0/activate/was/finding
or
https://qualysapi.qualys.com/qps/rest/3.0/activate/was/finding/<id>

Methods allowed: POST

Input

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

id (Integer)	ignoredReason (Keyword: FALSE_POSITIVE, RISK_ACCEPTED or NOT_APPLICABLE)
qid (Integer)	group (Keyword: XSS, SQL, INFO, PATH, CC, SSN_US or CUSTOM)
name (Text)	owasp.name (Text)
type (Keyword: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED)	owasp.code (Integer)
url (Text)	wasc.name (Text)
webapp.tags.id (Integer)	wasc.code (Integer)
webapp.tags.name (Text)	cwe.id (Integer)
status (Keyword: NEW, ACTIVE or REOPENED)	firstDetectedDate (Date)
Webapp.id (Integer)	lastDetectedDate (Date)
webapps.name (Text)	lastTestedDate (Date)
severity (Integer)	timesDetected (Integer)
ignoredDate (Date)	

Permissions

The WAS application must be enabled in the user's account
User must have "API Access" permission
Web application must be within the user's scope
User must have "Ignore Vulnerabilities" permission

Examples

Example 1: Activate all ignored findings (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/activate/was/finding"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>3</count>
  <data>
    <Finding>
      <id>1613225669</id>
    </Finding>
    <Finding>
      <id>1613255669</id>
    </Finding>
    <Finding>
      <id>1645195669</id>
    </Finding>
  </data>
</ServiceResponse>
```

Example 2: Activate specific finding (POST)

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/activate/was/finding/16
13255669"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Finding>
      <id>1613255669</id>
    </Finding>
  </data>
</ServiceResponse>
```

CHAPTER 10

Progressive Scanning

Qualys WAS 4.0 introduces Progressive Scanning to improve testing coverage and scheduling flexibility. Progressive Scanning is a limited availability feature, so if you're interested in becoming an early adopter please contact your Technical Account Manager or our Support Team. Learn more

We've made several updates to the WAS API to support Progressive Scanning - see below for all the details. Users will see these changes only when Progressive Scanning is enabled for their account.

Web Application API

Scan API

Schedule API

Scan Report

Web Application API

Schema: webapp.xsd

A new progressiveScanning element was added to the WebApp XML element. Expected format is a boolean.

```
<xs:complexType name="WebApp">
    <xs:all>
        <xs:element name="id" type="xs:long" minOccurs="0"/>
        ...
        <xs:element name="progressiveScanning" type="xs:boolean"
default="true" minOccurs="0"/>
```

CREATE / UPDATE web application

The user will be able to set progressiveScanning to true or false, if Progressive Scanning is enabled for the subscription. When Progressive Scanning is enabled for the subscription, if progressiveScanning option is not specified during CREATE request, by default the option will be enabled for the web application.

API request (create):

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/webapp/" <
file.xml
```

"file.xml" contains the request POST data:

```
<ServiceRequest>
  <data>
    <WebApp>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://mywebapp.com]]></url>
      <progressiveScanning>false</progressiveScanning>
    </WebApp>
  </data>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
```

```
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>1912949</id>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://mywebapp.com]]]></url>
...
            <scannerLocked>false</scannerLocked>
            <progressiveScanning>false</progressiveScanning>
...
```

XML output (error):

If Progressive Scanning is not enabled for the subscription, the progressiveScanning element cannot not be provided, otherwise an error will be returned.

```
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
    <responseCode>INVALID_REQUEST</responseCode>
    <responseErrorDetails>
        <errorMessage>Progressive scanning is not enabled in your
subscription.</errorMessage>
       <errorResolution>Please check with your account manager to
enable this option.</errorResolution>
    </responseErrorDetails>
</ServiceResponse>
```

GET web application

If Progressive Scanning is enabled for the subscription, the progressiveScanning element is displayed in GET call responses. If Progressive Scanning is not enabled for the subscription, the element is not included.

For all existing web applications created prior to WAS 4.0 the value will be set to TRUE by default.

API request:

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/webapp/323102"
```

XML output:

```
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebApp>
            <id>323102</id>
            <name>
                <![CDATA[MamboCMS]]>
            </name>
            <url>
<![CDATA[http://funkytown.acme01.acme.com/Forms/FormFields/temp/up
dated_web_app_name]]>
            </url>
...
            <scannerLocked>false</scannerLocked>
            <progressiveScanning>false</progressiveScanning>
...
```

Scan API

Schema: scan.xsd / wasscan.xsd

A new progressiveScanning element was added to the WasScan XML element. Expected format is a boolean.

```
<xs:complexType name="WasScan">
    <xs:all>
        <xs:element name="id" type="xs:long" minOccurs="0"/>
        ...
        <xs:element name="progressiveScanning" type="xs:boolean"
minOccurs="0"/>
```

LAUNCH scan

The user can set the progressiveScanning option to true or false for the vulnerability scan, if Progressive Scanning is enabled for the subscription. If the option is not set for a scan, the Progressive Scanning setting for the web application is used. Note this option is not supported for a discovery scan.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/launch/was/wasscan" <
file.xml
```

"file.xml" contains the request POST data:

```
<ServiceRequest>
  <data>
    <WasScan>
      <name>New WAS Vulnerability Scan launched from API</name>
      <type>VULNERABILITY</type>
      <target>
        <webApp>
            <id>323126</id>
        </webApp>
    <scannerAppliance>
       <type>EXTERNAL</type>
    </scannerAppliance>
      </target>
```

```
        <profile>
            <id>1021</id>
        </profile>
        <cancelAfterNHours>5</cancelAfterNHours>
         <progressiveScanning>false</progressiveScanning>
    </WasScan>
  </data>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>16954</id>
    </WasScan>
  </data>
</ServiceResponse>
```

XML output (error):

If Progressive Scanning is not enabled for the subscription, the progressiveScanning element cannot be provided, otherwise an error will be returned.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
    <responseCode>INVALID_REQUEST</responseCode>
    <responseErrorDetails>
        <errorMessage>Progressive scanning is not enabled in your
subscription.</errorMessage>
       <errorResolution>Please check with your account manager to
enable this option.</errorResolution>
    </responseErrorDetails>
</ServiceResponse>
```

GET scan

The progressiveScanning element will be included in the call response, if Progressive Scanning is enabled for the subscription. For all scans launched before this feature was enabled, the value "false" will be returned.

API request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/wasscan/31397"
XML output:
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WasScan>
            <id>31397</id>
            <name>
                <![CDATA[Relaunch Relaunch Web Application
Vulnerability Scan - 2014-08-13]]>
            </name>
            <reference>was/1413891468597.1792880</reference>
            <type>VULNERABILITY</type>
            <mode>ONDEMAND</mode>
            <progressiveScanning>true</progressiveScanning>
...
```

Schedule API

Schema: schedule.xsd / wasscanschedule.xsd

A new progressiveScanning element was added to the WasScanSchedule XML element.

DEFAULT	Use web application setting
ENABLED	Use progressive scanning for all web applications
DISABLED	Do not use progressive scanning for all web applications

```
<xs:simpleType name="ProgressiveScanningOption">
     <xs:restriction base="xs:string">
         <xs:enumeration value="DEFAULT"/>
         <xs:enumeration value="ENABLED"/>
         <xs:enumeration value="DISABLED"/>
     </xs:restriction>
 </xs:simpleType>

<xs:complexType name="WasScanSchedule">
    <xs:all>
        <xs:element name="id" type="xs:long" minOccurs="0"/>
        ...
        <xs:element name="progressiveScanning"
type="ProgressiveScanningOption" default="DEFAULT" minOccurs="0"/>
```

CREATE / UPDATE schedule

The user will be able to set progressiveScanning to ENABLED, DISABLED or DEFAULT, if progressiveScanning is enabled for the subscription. If this option is not set for a new schedule, the value DEFAULT is used.

API request (update):

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/wasscansched
ule/1688” < file.xml
```

“file.xml” contains the request POST data:

```
<ServiceRequest>
  <data>
```

```
    <WasScanSchedule>
      <id>1688</id>
      <progressiveScanning>ENABLED</progressiveScanning>
    </WasScanSchedule>
  </data>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>1688</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

XML output (error):

If Progressive Scanning is not enabled for the subscription, the progressiveScanning element cannot be provided, otherwise an error will be returned.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscanschedule.xsd">
    <responseCode>INVALID_REQUEST</responseCode>
    <responseErrorDetails>
        <errorMessage>Progressive scanning is not enabled in your
subscription.</errorMessage>
       <errorResolution>Please check with your account manager to
enable this option.</errorResolution>
    </responseErrorDetails>
</ServiceResponse>
```

GET schedule

The progressiveScanning element will be included in the call response, if Progressive Scanning is enabled for the subscription.

API request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/wasscanschedule
/8183"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscanschedule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WasScanSchedule>
            <id>8183</id>
            <name>
                <![CDATA[WASUI-3772 #3]]>
            </name>
...
            <progressiveScanning>ENABLED</progressiveScanning>
...
```

Scan Report

We've made enhancements to scan results data when Progressive Scanning is enabled for the subscription. When a vulnerability scan is processed, all scan findings will be associated with the consolidated finding for the web application. This occurs for every vulnerability scan, even if Progressive Scanning is not enabled for the scan. The consolidated finding is assigned a finding ID.

New elements in XML scan report

New elements appear in the XML scan report when Progressive Scanning is enabled for the subscription as follows.

The new DETECTION_ID element (optional) containing the web application finding ID appears following these elements. (If the finding ID is unavailable the element is not included).

```
RESULTS/VULNERABILITY_LIST/VULNERABILITY/DETECTION_ID
RESULTS/SENSITIVE_CONTENT_LIST/SENSITIVE_CONTENT/DETECTION_ID
RESULTS/INFORMATION_GATHERED_LIST/INFORMATION_GATHERED/DETECTION_I
D
```

The new PROGRESSIVE_SCANNING element is included in the Appendix. This is set to true if the scan used Progressive Scanning, or false.

Sample XML scan report

```
...
<RESULTS>
        <VULNERABILITY_LIST>
            <VULNERABILITY>
                <ID>76681</ID>
                <DETECTION_ID>7098</DETECTION_ID>
                <QID>150004</QID>
<URL><![CDATA[https://10.10.26.238/../../boq/protected/mime/defaul
tPage.pdf.test1]]></URL>
                <AUTHENTICATION>Not Required</AUTHENTICATION>
                <STATUS>NEW</STATUS>
                <FIRST_TIME_DETECTED>2014-10-
07T18:09:52Z</FIRST_TIME_DETECTED>
                <LAST_TIME_DETECTED>2014-10-
07T18:09:52Z</LAST_TIME_DETECTED>
                <LAST_TIME_TESTED>2014-10-
07T18:09:52Z</LAST_TIME_TESTED>
```

```
                <TIMES_DETECTED>1</TIMES_DETECTED>
                <PAYLOADS>
                    <PAYLOAD>
                        <NUM>1</NUM>
<PAYLOAD><![CDATA[/../..@PATH@@FILENAME@.@EXTENSION@.test1]]></PAY
LOAD>
                        <REQUEST>
                            <METHOD>GET</METHOD>
<URL><![CDATA[https://10.10.26.238/../../boq/protected/mime/defaul
tPage.pdf.test1]]></URL>
                            <HEADERS>
                                <HEADER KEY="Referer">
<![CDATA[http://10.10.26.238/]]> </HEADER>
                                <HEADER KEY="Cookie"><![CDATA[
cookie3=cookiethree; cookie2=cookietwo; cookie1=cookieone;
PHPSESSID=346e3909391063e3449e203efaefa224;
]]></HEADER>
                            </HEADERS>
                        </REQUEST>
                        <RESPONSE>
                            <CONTENTS
base64="true"><![CDATA[SFRUUC8xLjEgMjAwIE9L]]></CONTENTS>
                        </RESPONSE>
                    </PAYLOAD>
                </PAYLOADS>
            </VULNERABILITY>

...

<SENSITIVE_CONTENT>
                <ID>76671</ID>
                <DETECTION_ID>2304</DETECTION_ID>
                <QID>150022</QID>
<URL><![CDATA[http://10.10.26.238/boq/acct/other/WillProduce500Err
or.php]]></URL>
                <CONTENT>"%20onEvent=@REQUESTID@%20</CONTENT>
                <PARAM><![CDATA[referer]]></PARAM>
                <ACCESS_PATH>
                    <URL><![CDATA[http://10.10.26.238/]]></URL>
<URL><![CDATA[http://10.10.26.238/boq/tou.html?s=1]]></URL>

<URL><![CDATA[http://10.10.26.238/boq/acct/]]></URL>
```

```
<URL><![CDATA[http://10.10.26.238/boq/acct/other/]]></URL>
                </ACCESS_PATH>
                <AUTHENTICATED>Not Required</AUTHENTICATED>
                <STATUS>NEW</STATUS>
                <FIRST_TIME_DETECTED>2014-10-
07T18:09:52Z</FIRST_TIME_DETECTED>
                <LAST_TIME_DETECTED>2014-10-
07T18:09:52Z</LAST_TIME_DETECTED>
                <LAST_TIME_TESTED>2014-10-
07T18:09:52Z</LAST_TIME_TESTED>
                <TIMES_DETECTED>1</TIMES_DETECTED>
                <PAYLOADS>
                    <PAYLOAD>
                        <NUM>1</NUM>
                        <PAYLOAD><![CDATA[%00]]></PAYLOAD>
                        <REQUEST>
                            <METHOD>GET</METHOD>
                            <URL>
<![CDATA[http://10.10.26.238/boq/acct/other/WillProduce500Error.ph
p]]></URL>
                            <HEADERS>
                                <HEADER KEY="Referer"><![CDATA[
http://localhost/%00]]></HEADER>
                                <HEADER KEY="Cookie"><![CDATA[
cookie3=cookiethree; cookie2=cookietwo; cookie1=cookieone;
PHPSESSID=346e3909391063e3449e203efaefa224;
]]></HEADER>
                            </HEADERS>
                        </REQUEST>
                        <RESPONSE>
                            <CONTENTS
base64="true"><![CDATA[PEhUTUw-
CiAgPEhFQUQ-CiAgPC9IRUFEPgogIDxCT0RZPgogIDwvQk9EWT4KPC9IVE1MPg]]><
/CONTENTS>
                        </RESPONSE>
                    </PAYLOAD>
                </PAYLOADS>
            </SENSITIVE_CONTENT>

...

<APPENDIX>
```

```
        <SCAN_LIST>
            <SCAN>
               <NAME><![CDATA[VULN SCAN - 10.10.26.238 #2]]></NAME>
                 <REFERENCE>was/1412697001370.1789435</REFERENCE>
                 <START_DATE>2014-10-07T17:50:01Z</START_DATE>
                 <END_DATE>2014-10-07T18:09:52Z</END_DATE>
                 <MODE>Vulnerability</MODE>
                 <PROGRESSIVE_SCANNING>false</PROGRESSIVE_SCANNING>
                 <TYPE>Manual</TYPE>
                 <WEB_APPLICATION><![CDATA[10.10.26.238 - Good
One!]]></WEB_APPLICATION>
                 <AUTHENTICATION_RECORD><![CDATA[AAA
(ID=2005,10.10.26.238 - Good One!)]]></AUTHENTICATION_RECORD>
                 <PROFILE><![CDATA[CC/SSN]]></PROFILE>
                 <SCANNER>External (IP: 10.10.21.160, Scanner:
7.9.40-1, WAS: 3.6.46-1, Signatures: 2.2.813-1)</SCANNER>
                 <STATUS>Finished</STATUS>

<AUTHENTICATION_STATUS>Partial</AUTHENTICATION_STATUS>
            </SCAN>
```

APPENDIX A

Error Messages

This appendix describes the types of error messages returned from WAS API requests.

Sample Messages: Elements

Sample Messages: Criteria

Sample Messages: Authorization

Sample Messages: Report Storage Limit

Sample Messages: Elements

Sample messages for element errors are shown below.

Error Message	Resolution
Element Validation	
url: Invalid URL format (<value>).	URL format must be as follows: http://<baseUrl>/rest/3.0/?parameters
<scope>: Invalid value (<value>).	Element must be set to one of these values: ALL, LIMIT, SUBDOMAIN or DOMAINS.
domains: Element is required when scope is set to: DOMAINS.	Specify the domains to include in the web application scope in the "domains" element.
subDomain: Element is required when scope is set to: SUBDOMAIN.	Specify the subdomains to include in the web application scope in the "subDomain" element.
subDomain: Invalid domain name format (<value>).	Use following format in the "subDomain" element: .my.domain.suffix (must start with a dot)
useRobots: Invalid value (<value>).	Element "userRobots" must be set to one of these values: IGNORE, ADD_PATHS, BLACKLIST.
Url: Element is required	Element "Url" is required.
uris.<field>: Invalid URL format (<value>).	For the uri.<field> sub element, specify a URL like http://domain.name/base/url/?parameters
uris.<field>: Length of the field must not be greater than 2048 characters. (<value>).	For the uri.<field> sub element, the maximum field length is 2048 characters.
Domain: Element is required	The domain element must be provided.
Domain: Invalid host name format (<value>).	Use following format for value in the "Domain" element: www.my.domain.example.
Length of all domains cannot exceed 2048 characters.	The list of all domains in the web application cannot exceed 2048 characters.
Attribute.category: Element is required.	The element Attribute.category is required.
Attribute.category: Invalid value (<value>).	Element Attribute.category must be set to one of these values: Business Function, Business Location, Business Description.
Attribute.value: Element is required.	Provide a value for the attribute in the Attribute.value element: function, location or description.
The attribute length cannot be greater than 64 characters.	The value for this attribute cannot exceed 64 characters.
The attribute length cannot be greater than 2048 characters.	The value for this attribute cannot exceed 2048 characters.

Error Message	Resolution
<element>: Element must not be set.	This element does not apply to this request.
set: Element must contain at least one child.	The set element requires at least one sub element.
At least one of the following elements must be set: set, add, remove.	This request requires at least one of these elements: set, add or remove.
headers: Length of all headers cannot exceed 2048 characters.	The values of all headers cannot exceed 2048 characters.
At least one of the following elements must be set: set, add, remove.	For an "update" request you must set at least one of these elements: set, add or remove.
UrlEntry: Element is required.	The element UrlEntry must be provided.
UrlEntry: Invalid URL format (value).	Specify a URL like http://domain.name/base/url/?parameters
<parent>: Length of all [URLs, regular expressions] cannot exceed 2048 characters	The list of entries for a given type shall not exceed 2048 characters.
UrlEntry: Only regular expressions are accepted for this element.	You must provide regular expressions for the element postDataBlackList.
tags.<element>: Element must not be set.	The tags element does not apply for this request
tags.set: Element must contain at least one child.	At least one sub element must be provided for the element tag.set.
Tag.id: Element is required.	Provide a value for the element Tag.id
Tag.id: Invalid value (value).	Value must be an integer set at least to 1.
Tag: Tag specified by ID <id> does not exist or is not available.	Provide a value for the element id that corresponds to a valid tag.

Sample Messages: Criteria

Sample messages for errors related to criteria are shown below.

Error Message	Resolution
Element Validation	
Criteria: Field is required.	Specify the name of the criteria to search against.
Criteria: Invalid criteria (<field name>).	Please search against one of the following criteria: %s.
Criteria: Invalid operator for criteria '<field>' (<operator>).	Allowed operations for this criteria are: %s.
Criteria: Value is required for criteria '<field>'.	Specify a value for a field name for search criteria.
Criteria: Invalid value format for criteria '<field>': <value>.	Boolean (true, false). Date and Time in UTC format Enumeration (allowed options separated by comma). Other: Specify criteria value(s) as <type>.

Sample Messages: Authorization

Sample messages related to authorization errors are shown below.

Error Message	Resolution
Element Validation	
You are not authorized to access the application through the API.	You must be granted the API Access permission in your roles and scopes.
You do not have access to module Web Application Scanning required by this API.	Please contact your account manager to have WAS enabled in your subscription.
No data shall be passed for this operation.	The POST request does not specify a data element.
User is not authorized to perform this operation on specified object(s).	You must be granted access to these objects in your user scope.
Operation %s does not support search filters.	Do not provide search filers for this operation.
Quota of web application has been exceeded.	Please check with your account manager to purchase new applications.

Sample Messages: Report Storage Limit

A sample message related to report storage limit is shown below.

Error Message	Resolution
Element Validation	
Your [subscription|user] storage limit of <NB> Mb has been reached.	Delete existing reports and try again.

APPENDIX B

WAS Findings in XML Reports

Findings in all WAS reports in XML format are Base64 encoded starting with WAS version 3.1 and later. Findings include vulnerability detections, information gathered and sensitive content.

Did you build clients using WAS version 3.0 or earlier? If yes, please update your clients so that WAS findings data is processed accurately.

Tell me about Base64 encoded findings

All findings reported for scan and web applications are base64 encoded in XML. This includes:

- Actual contents of the response

- If evidence in response is highlighted, the evidence contents

- Information gathered data

Base64 encoded data usually will have the attribute set to “base64=true”. For example:

```
<FINDING>
  <PAYLOAD><![CDATA[uid=%00%3Cscript%3E_q%3Drandom(X157105156Y1
Z)%3C%2Fscript
%3E]]></PAYLOAD>
  <RESULT
base64="true"><![CDATA[Cl9mZWVkKCgKCgpbCiI=]]></RESULT>
</FINDING>
```

If the "base64=true attribute" is not set, the value will be in plain text. For example:

```
<FINDING>
  <PAYLOAD><![CDATA[uid=%00%3Cscript%3E_q%3Drandom(X157105156Y1Z)%
3C%2Fscript
%3E]]></PAYLOAD>
  <RESULT><![CDATA[_feed(("]]></RESULT>
</FINDING>
```

Which WAS reports show findings?

- WAS v3 Scan Results

- Web Application Report

- Web Application Scan Report

WAS v3 Scan Results

Vulnerability and Sensitive Content findings
WasScan/vulns/list/WasScanVuln/instances/list/WasScanVulnInstance/payloads/list/WasScanVulnPayload/result
WasScan/sensitiveContents/list/WasScanSensitiveContent/instances/list/WasScanSensitiveContentInstance/payloads/list/WasScanSensitiveContentPayload/result

Sample WAS v3 Scan Results XML:

```
<WasScanVuln>
  <qid>150001</qid>
  <title><![CDATA[Reflected Cross-Site Scripting (XSS)
Vulnerabilities]]></title>
<uri><![CDATA[http://myuri.apps.com/613460625329/feed.gtl?uid=%22'
%3E%3Cqss%20a%3DX157
105156Y1Z%3E]]></uri>
  <param>uid</param>
  <instances>
    <count>1</count>
    <list>
      <WasScanVulnInstance>
        <authenticated>false</authenticated>
        <payloads>
          <count>4</count>
          <list>
```

```
            <WasScanVulnPayload>
<payload><![CDATA[uid=%00%3Cscript%3E_q%3Drandom(X157105156Y1Z)%3C
%2Fscript%3E]]>
            </payload>
            <result base64="true">
<![CDATA[Cl9mZWVkKCgKCgpbCiI]]></result>
            </WasScanVulnPayload>
            <WasScanVulnPayload>
<payload><![CDATA[uid=%22'%3E%3Cqss%20a%3DX157105156Y1Z%3E]]></pay
load>
            <result base64="true">
<![CDATA[Cl9mZWVkKCgKCgpbCiIiJyZndDsmbHQ7cXNzIGE9WDE1NzEwNTE1NlkxW
iZndDsiCgpdCgoKCikpCg]]></result>
            </WasScanVulnPayload>
            <WasScanVulnPayload>
<payload><![CDATA[uid=%00%3Cscript%3E_q%3Drandom(X157201836Y1Z)%3C
%2Fscript%3E]]>
            </payload>
            <result
base64="true"><![CDATA[Cl9mZWVkKCgKCgpbCiI]]></result>
            </WasScanVulnPayload>
            <WasScanVulnPayload>
<payload><![CDATA[uid=%22'%3E%3Cqss%20a%3DX157201836Y1Z%3E]]></pay
load>
            <result base64="true">
<![CDATA[Cl9mZWVkKCgKCgpbCiIiJyZndDsmbHQ7cXNzIGE9WDE1NzIwMTgzNlkxW
iZndDsiCgpdCgoKCikpCg]]></result>
            </WasScanVulnPayload>
          </list>
        </payloads>
      </WasScanVulnInstance>
    </list>
  </instances>
</WasScanVuln>
```

Information Gathered findings
WasScan/igs/list/WasScanIg/data

Sample WAS v3 Scan Results XML:

```
<INFO>
  <QID>150044</QID>
```

```
  <TITLE><![CDATA[Login Form Is Not Submitted Via HTTPS]]></TITLE>
  <RESULT base64="true">
<![CDATA[RGVmYXVsdCBmb3JtIGFjdGlvbiBkb2VzIG5vdCBzdWJtaXQgdmlhIFNTT
DogaHR0cDovL2dvb2ds
ZS1ncnV5ZXJlLmFwcHNwb3QuY29tLzYxMzQ2MDYyNTMyOS9sb2dpbgo=]]></RESUL
T>
</INFO>
```

Web Application Report

Vulnerability and Sensitive Content findings
WAS_WEBAPP_REPORT/RESULTS/WEB_APPLICATION/VULNERABILITY_LIST/VULNERABILITY/PAYLOADS/PAYLOAD/RESPONSE/CONTENTS
WAS_WEBAPP_REPORT/RESULTS/WEB_APPLICATION/SENSITIVE_CONTENT_LIST/SENSITIVE_CONTENT/PAYLOADS/PAYLOAD/RESPONSE/CONTENTS
WAS_WEBAPP_REPORT/RESULTS/WEB_APPLICATION/VULNERABILITY_LIST/VULNERABILITY/PAYLOADS/PAYLOAD/RESPONSE/EVIDENCE
WAS_WEBAPP_REPORT/RESULTS/WEB_APPLICATION/SENSITIVE_CONTENT_LIST/SENSITIVE_CONTENT/PAYLOADS/PAYLOAD/RESPONSE/EVIDENCE

Sample Web Application Report XML:

```
<VULNERABILITY>
  <ID>5943</ID>
  <QID>150001</QID>
<URL><![CDATA[http://myuri.apps.com/app/xss/0/1/0/xss.php?s='%20on
Event%3dX146470180Y1Z%20]]></URL>
  <PARAM><![CDATA[s]]></PARAM>
  <AUTHENTICATION>Not Required</AUTHENTICATION>
  <STATUS>NEW</STATUS>
  <FIRST_TIME_DETECTED>2011-12-30T09:57:39Z</FIRST_TIME_DETECTED>
  <LAST_TIME_DETECTED>2011-12-30T09:57:39Z</LAST_TIME_DETECTED>
  <LAST_TIME_TESTED>2011-12-30T09:57:39Z</LAST_TIME_TESTED>
  <TIMES_DETECTED>1</TIMES_DETECTED>
  <PAYLOADS>
    <PAYLOAD>
      <NUM>1</NUM>
<PAYLOAD><![CDATA[s='%20onEvent%3dX146470180Y1Z%20]]></PAYLOAD>
      <REQUEST/>
      <RESPONSE>
        <CONTENTS
```

```
base64="true"><![CDATA[bGQiJmd0OyZsdDsmbHQ7L3NwYW4mZ3Q7ID0mZ3Q7ICZ
sdDtzcGFuIGNsYXNzPSJib2xkIiZndDsmYW1wO2x0OyZsdDsvc3BhbiZndDsmbHQ7Y
nImZ3Q7CiZsdDsvZGl2Jmd0OwombHQ7L2RpdiZndDsKJmx0O2JyJmd0OwombHQ7ZGl
2IGNsYXNzPSJwYXlsb2FkcyImZ3Q7Ck91dHB1dCBmcm9tIHJlcXVlc3QgJmx0O3NwY
W4gY2xhc3M9ImJvbGQiJmd0Oy9jYXNzaXVtL3hzcy5waHA/dmFyaWFudD0wJmFtcDt
xcz0xJmFtcDtmPTAmYW1wO3M9JyUyMG9uRXZlbnQlM2RYMTQ2NDcwMTgwWTFaJTIwJ
mx0Oy9zcGFuJmd0OwombHQ7YnImZ3Q7CiZsdDthIGhyZWY9J1wnIG9uRXZlbnQ9WDE
0NjQ3MDE4MFkxWiAnJmd0O3NhbXBsZSBsaW5rJmx0Oy9hJmd0OwombHQ7L2RpdiZnd
DsKJmx0O3NjcmlwdCZndDttYWluKCkmbHQ7L3NjcmlwdCZndDsKJmx0Oy9ib2R5Jmd
0OwombHQ7L2h0bWwmZ3Q7]]></CONTENTS>
      </RESPONSE>
    </PAYLOAD>
  </PAYLOADS>
    <IGNORED>false</IGNORED>
</VULNERABILITY>
```

Information Gathered findings
WAS_WEBAPP_REPORT/RESULTS/WEB_APPLICATION/INFORMATION_GATHERED_LIST/INFORMATION_GATHERED/DATA

Sample Web Application Report XML:

```
<INFORMATION_GATHERED_LIST>
  <INFORMATION_GATHERED>
    <ID>1529</ID>
    <QID>6</QID>
    <FIRST_TIME_DETECTED>2011-12-
30T09:57:39Z</FIRST_TIME_DETECTED>
    <LAST_TIME_DETECTED>2011-12-30T09:57:39Z</LAST_TIME_DETECTED>
    <LAST_TIME_TESTED>2011-12-30T09:57:39Z</LAST_TIME_TESTED>
<DATA
base64="true"><![CDATA[I3RhYmxlCklQX2FkZHJlc3MgSG9zdF9uYW1lCgoxMC4
xMC4yNi43NyBmdW5reXR vd24udnVsbi5x
YS5xdWFseXMuY29tCg==]]></DATA>
  </INFORMATION_GATHERED>
  <INFORMATION_GATHERED>
    <ID>1532</ID>
    <QID>150031</QID>
    <FIRST_TIME_DETECTED>2011-12-
30T09:57:39Z</FIRST_TIME_DETECTED>
    <LAST_TIME_DETECTED>2011-12-30T09:57:39Z</LAST_TIME_DETECTED>
    <LAST_TIME_TESTED>2011-12-30T09:57:39Z</LAST_TIME_TESTED>
```

```
<DATA
base64="true"><![CDATA[VGltZW91dCByZWFjaGVkIGluIElQQyBjb25uZWN0aW9
uIHRvIFdlYktpdC4gSmF
2YVNjcmlwdCBz
dXBwb3J0IGRpc2FibGVkIGluOmVQaGFzZUNyYXdsCkNyYXdsIGNvbXBsZXRlZCB3aX
RoIFdlYktp dC4K]]></DATA>
</INFORMATION_GATHERED>
```

Web Application Scan Report

Vulnerability and Sensitive Content findings
WAS_SCAN_REPORT/RESULTS/VULNERABILITY_LIST/VULNERABILITY/PAYLOADS/PAYLOAD/RESPONSE/CONTENTS
WAS_SCAN_REPORT/RESULTS/SENSITIVE_CONTENT_LIST/SENSITIVE_CONTENT/PAYLOADS/PAYLOAD/RESPONSE/CONTENTS
WAS_SCAN_REPORT/RESULTS/VULNERABILITY_LIST/VULNERABILITY/PAYLOADS/PAYLOAD/RESPONSE/EVIDENCE
WAS_SCAN_REPORT/RESULTS/SENSITIVE_CONTENT_LIST/SENSITIVE_CONTENT/PAYLOADS/PAYLOAD/RESPONSE/EVIDENCE
Information Gathered findings
WAS_SCAN_REPORT/RESULTS/INFORMATION_GATHERED_LIST/INFORMATION_GATHERED/DATA